

# A Framework for Large-Force Task Planning of Mobile and Redundant Manipulators

**Evangelos Papadopoulos\***

*Department of Mechanical Engineering  
National Technical University of Athens  
Athens 157 73, Greece  
e-mail: egpapado@central.ntua.gr*

**Yves Gonthier**

*Centre de recherche informatique de Montreal  
Montréal, Que. H3A 2N4, Canada*

Received February 19, 1997; revised September 20, 1998;  
accepted October 31, 1998

A framework tackling the problem of large wrench application using robotic systems with limited force or torque actuators is presented. It is shown that such systems can apply a wrench to a limited set of Cartesian locations called force workspace (FW), and its force capabilities are improved by employing base mobility and redundancy. An efficient numerical algorithm based on  $2^n$ -tree decomposition of Cartesian space is designed to generate FW. Based on the FW generation algorithm, a planning method is presented resulting in proper base positioning relative to large-force quasistatic tasks. Additionally, the case of tasks requiring application of a wrench along a given path is considered. Task workspace, the set of Cartesian space locations that are feasible starting positions for such tasks, is shown to be a subset of FW. This workspace is used for identifying proper base or task positions guaranteeing task execution along desired paths. Finally, to plan redundant manipulator postures during large-force-tasks, a new method based on a min-max optimization scheme is developed. Unlike norm-based methods, this method guarantees no actuator capabilities are exceeded, and force or torque of the most loaded joint is minimized. Illustrative examples are given demonstrating validity and usefulness of the proposed framework. © 1999 John Wiley & Sons, Inc.

\* To whom all correspondence should be addressed.

## I. INTRODUCTION

Animals and humans can develop and apply large forces, compared to their muscle force or torque capabilities. During a task requiring application of large forces, the body is positioned at a location near the task and an appropriate arm posture is assumed. The body position, or arm posture may change during the execution of the task, and both mobility and redundancy are used efficiently. On the other hand, robotic manipulators exhibit limited force capabilities, even in static or quasistatic tasks. This issue becomes very important in mobile applications of robotic systems, where typically development of large forces is expected. In these applications, the position of a mobile base of a system can be relocated with respect to a task, adding redundancy to the system. In cases where repetitious tasks are being planned, the robot can be positioned initially such that its posture is optimal for the given force task. In space, highly redundant systems are being built to operate in a gravityless environment. The application of large forces in this environment can be problematic given the fact that actuators are typically small due to weight restrictions, and to the lack of the need to support the weight of the system.

Actuator limitations have been considered in studies of time optimal motion planning, and in resolving manipulator redundancy in motion control.<sup>1,2</sup> The force distribution problem in multilimb systems has been studied using linear programming techniques, in conjunction to energy and load balancing performance indexes.<sup>3</sup> The necessary and sufficient conditions for applying a force to the environment were presented in ref. 4. Posture control in motion or force tasks has been considered using velocity and force ellipsoids, and a task compatibility measure.<sup>5</sup> The direction and magnitude of maximum force and torque that can be applied at some given end-effector location has been studied to provide a basis for the task planning for force control of multiple cooperating robot arms.<sup>6</sup> Weight-lifting under limited torque capabilities was analyzed in refs. 7–9. In these works, it is assumed that a weight is moved from a low to a high fixed point, while the path is not specified. Norm-based criteria are used to find the optimum path for lifting a weight between two extreme points,<sup>7</sup> while integral norm-based criteria are employed in ref. 9. The effects of limited actuator capabilities on the force output of multilimbed systems were introduced first in ref. 10. A configuration-space force workspace

was defined and used to plan motions of such systems without violating actuation and joint limits, or frictional constraints.<sup>10</sup> Redundancy resolution criteria were introduced and evaluated based on desired motion or force task requirements.<sup>11</sup>

This article analyzes the application of large forces and torques, i.e., wrenches, by robotic systems with limited force or torque actuators. Such a system may be able to apply a wrench in some configurations only; therefore its useful force workspace (FW) is limited. However, its force capabilities can be improved by employing base mobility and redundancy. An efficient numerical algorithm based on the  $2^n$ -tree decomposition of the Cartesian space is designed to generate the force workspace. Based on this algorithm, a planning method is presented which results in proper base positioning relative to large-force quasistatic tasks. If this method results in no solution, then the task is either unfeasible, or has to be divided into two subtasks whose completion requires base motion during task execution.

In addition, the case of tasks requiring the application of a wrench along a specified path is considered. The task workspace, the set of Cartesian space locations that are feasible starting positions for such tasks, is shown to be a subset of the force workspace. This workspace can be used for identifying proper base or task positions that will guarantee task execution along desired paths. Finally, to plan redundant manipulator postures during large force-tasks, a new method based on a min-max optimization method is developed and compared to an existing norm-based one. It is shown that the latter does not guarantee that actuator limits are not exceeded, and that they can indeed be exceeded. In contrast, the min-max optimization method guarantees that all joint forces or torques do not exceed their limits, and that each actuator "suffers" the least possible, resulting in effective actuator use, and maximization of a system's force capabilities. Illustrative examples are given that demonstrate the validity and usefulness of the proposed framework.

## II. FORCE WORKSPACE CONCEPTS

In this article we focus on tasks which require the application of large wrenches on the environment. For example, such tasks include holding or moving large payloads, pushing heavy containers in warehouse operations, or removing of orbital removable

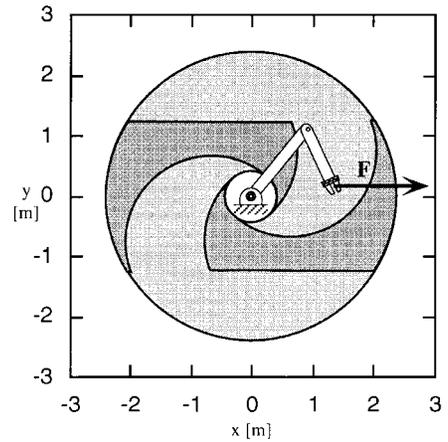
units (ORU) during contingency operations in space. Although manipulators may differ greatly in their nature, they are all subject to the physical limitations of their actuators, and they all operate in the same physical world most tasks are defined in: the Cartesian space. However, the effect of these limitations is not the same in all workspace locations. For example, a manipulator may be able to lift a large load when located close to its base, but not otherwise.

If we consider all possible manipulator configurations  $\mathbf{q}$  at which the end-effector of a robotic system can apply a given wrench  $\mathbf{F}$ , and map these to a set of Cartesian locations defined by the same property, then the union of all such locations defines a manipulator's *force workspace*. In other words, the force workspace (FW) includes all Cartesian space locations at which a robotic system can apply a given wrench without exceeding actuator limits. Note that as the kinematic workspace is reduced by the existence of joint limits, the FW is affected by actuator force or torque limits. If a desired wrench  $\mathbf{F}$  is relatively small, then all actuators are able to generate the required forces or torques at each joint, such that the end-effector is applying  $\mathbf{F}$  to the environment. In such a case, the FW is identical to the kinematic workspace. However, as  $\mathbf{F}$  increases, this workspace reduces gradually and may even vanish, i.e., there may be no points in the kinematic workspace for which the end-effector of the manipulator is able to apply  $\mathbf{F}$ .

To demonstrate this notion, a simple two-link planar manipulator is studied. The link lengths are  $l_1 = 1.4$  m and  $l_2 = 1.0$  m and actuator torque limits are  $\tau_{1,\max} = 10$  Nm and  $\tau_{2,\max} = 6$  Nm. For simplicity, the second joint is restricted to positive angles and the effect of gravity has been neglected. As depicted in Figure 1, when the force is very small, here 1 N along the positive  $x$  axis direction, the end-effector can apply it at any point in its reachable workspace. However, when the force is increased to 8 N, the available FW is reduced significantly.

### Task Positioning

Finding the FW for a system as a function of the end-effector wrench allows planning of force-tasks. For example, a large force can be applied to an object if this object is located in the corresponding FW. If it is not, as is the case for the manipulator configuration shown in Figure 1, then the calculation of this workspace suggests locations at which



**Figure 1.** Workspace regions where force  $F$  can be applied. Small  $F$  (light gray and dark gray), large  $F$  (dark gray).

the object must be moved prior to the execution of the force-task; it indicates *valid positions* for the task. Also, if a manipulator is required to apply a sequence of wrenches to an object, we can use the force workspace to determine a proper location for this object so that all the wrenches can be applied by the manipulator, without exceeding its actuator force or torque limits. The possible locations will be in the *intersection* of the force workspaces for each of the wrenches to be applied by the manipulator.

### Mobile Manipulator Base Positioning

Sometimes there is no freedom in choosing the position for a task, but it may be possible to relocate the base of the robotic system. For example, assume that the task for a robotic system is to lift some heavy object off the ground. The task may be unfeasible, if that object is located far from the robotic system. However, placing the manipulator closer to the task would load its actuators less and allow object lifting.

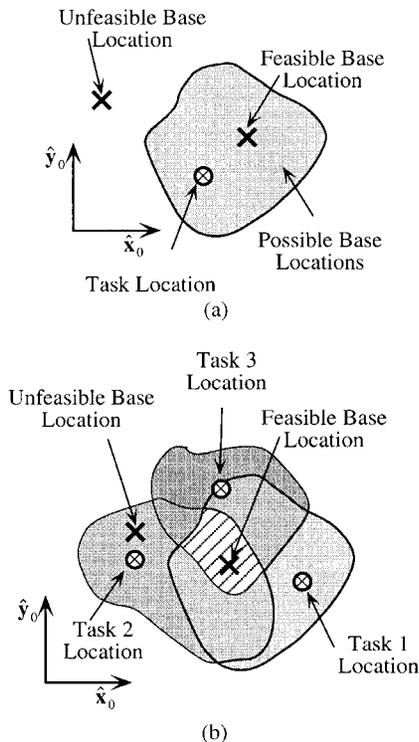
A base position is called *feasible* when the manipulator is able to execute a task successfully after its base is moved to this location. The FW can be used to determine the feasible base positions of a robotic system for a given task. Since the FW is defined in the Cartesian space, the following equation is true for all the end-effector locations inside the FW,

$$\mathbf{p}_{EE}^0 = \mathbf{p}_B^0 + \mathbf{p}_{FW}^B \quad (1)$$

where  $\mathbf{p}_{EE}^0$  is the end-effector position vector with respect to the world frame 0,  $\mathbf{p}_B^0$  is the base position of the robotic system with respect to the world frame, and  $\mathbf{p}_{FW}^B$  is a position vector that locates points of the FW with respect to the base position. Now, if the task to be executed is located at  $\mathbf{p}_T^0$ , then setting  $\mathbf{p}_{EE}^0 = \mathbf{p}_T^0$  in Eq. (1) results in

$$\mathbf{p}_B^0 = \mathbf{p}_T^0 - \mathbf{p}_{FW}^B \quad (2)$$

Equation (2) provides the means of finding base positions for which the task is located inside the force workspace of the system; hence, where the force task can be executed, see Figure 2a. As in the previous section, this concept can be extended to the case where the robotic system has to execute a number of tasks. In this case, we must generate a set of feasible base positions for each of the tasks. Note that the tasks do not need to be spatially coincident. Similarly to the previous section, the feasible base positions for a set of tasks will belong to the *intersection* between all individual base positions for each of the tasks. This concept is illustrated in Figure 2b.



**Figure 2.** Base positioning using the force workspace. (a) Single task, (b) multiple tasks.

### III. FORCE WORKSPACE GENERATION

When describing a volume of arbitrary shape, several methods are available. To describe it accurately, i.e., to know its exact shape, an analytical description is needed. However, in the case of the FW, such a description is not always available, and in fact, it can be found in simple cases only. Another approach is to approximate the shape of the workspace using numerical methods. A simple method to identify a space characterized by a given property, is to decompose it into a number of equally sized *elements*, and test each element to determine if it is part of this space or not. Here, the quality of the approximation depends on the *resolution*,  $r$ , of the approximation. As each element requires some calculations to determine whether or not it is part of the space, the number of calculations is of the order  $O(r^n)$ , where  $n$  is the dimension of the space. It is easy to see that such a method is not very efficient, especially for high  $n$ . The  $2^n$ -tree decomposition of the free space, as proposed by Paden, Mees, and Fisher is a more efficient technique.<sup>12</sup>

The  $2^n$ -tree is a variable resolution data structure. To apply this method, the Cartesian space is divided into a number of cells that can vary in size and correspond to the nodes in the tree. Each cell is a hyper cube and its dimension is that of the space it belongs to: a square for a two-dimensional space (quadtree), or a cube for a three-dimensional space (octree). The root node of the tree represents the entire Cartesian space under study. To generate the  $2^n$ -tree, a test function must be set up to determine the feasibility of each node; in our case this function tests whether a cell is part of the FW or not. Each node can have one of three states: either it is entirely *feasible* (it is contained in the FW), entirely *unfeasible* (outside the FW) or partially feasible (partially in the FW), in which case it is labeled *mixed*. If a node is feasible, it is added to the tree, if it is unfeasible, it is discarded, and if it is mixed we cannot add it nor discard it: we split the cell that corresponds to this node into  $2^n$  cells. Each new cell must then be tested to determine its feasibility, and the process continues. Samet shows that using this representation, the maximum number of nodes, which is proportional to the number of calculations, is of the order  $O(r^{n-1})$ .<sup>13</sup>

To determine the feasibility of a cell, we need to ensure that all of its points are valid locations for the end-effector, or equivalently, that none of the required actuator forces or torques exceed actuator capabilities. To this end, these actuator forces or

torques must be determined. Depending on the nature of the robotic system, this evaluation may be quite involved and require a significant amount of calculations.

To avoid calculating a system's actuator forces or torques for every location in the root cell of the Cartesian space, the algorithm displayed in Figure 3 was designed such that cells obviously not part of the FW are eliminated. First, the search for FW cells is restricted to those that are located entirely within the kinematic workspace of a robotic system. Evidently, if the manipulator cannot reach a point, it will not be able to apply the desired wrench. So the cells are first screened for *kinematic* feasibility by the test **TestKW**, see Figure 3.

Next, we consider manipulators for which the first joint is a revolute joint with its axis of rotation parallel to the direction of gravity. Note that this class includes the majority of robotic systems in use today. For these manipulators, we can reduce even further the volume to be searched. Because of the orientation of the first joint axis, the weight of the other links does not affect its actuator. When the end-effector of such a robotic system applies to its

environment a wrench  $\mathbf{F} = [\mathbf{f}^T \mathbf{n}^T]^T$  statically or quasistatically, where  $\mathbf{f}$  is the applied force and  $\mathbf{n}$  is the applied moment, the torque applied at the first joint is simply,

$$\tau_1 = (\mathbf{p}_{EE} \times \mathbf{f} + \mathbf{n}) \cdot \hat{\mathbf{z}}_1 \quad (3)$$

where  $\mathbf{p}_{EE}$  is the position vector of the end-effector with respect to the origin of frame 1 and  $\hat{\mathbf{z}}_1$  a unit vector along the first joint axis. For brevity, the superscript in  $\mathbf{p}_{EE}^1$  was omitted. Let us assume further that the first joint actuator has limited torque capabilities such that

$$-\tau_{1,\max} \leq \tau_1 \leq \tau_{1,\max} \quad (4)$$

From Eqs. (3) and (4) we find a condition relating the position of the end-effector to the maximum actuator torque of the first joint,

$$|(\mathbf{p}_{EE} \times \mathbf{f} + \mathbf{n}) \cdot \hat{\mathbf{z}}_1| \leq \tau_{1,\max} \quad (5)$$

Hence, any point in a feasible cell must verify Eq. (5). We call this the *first joint actuator constraint* feasibility test (**TestFJAC**). Using this test the search region can be narrowed even further. However, we must still determine if each node cell of the remaining volume is part of the FW or not, i.e., its feasibility.

To determine the feasibility of a node cell, we first set up a *point* feasibility criterion (PFC) that determines the feasibility of a single end-effector location (or *point*) in the Cartesian space. Assuming that a robotic system with  $N$  joint actuators has limited force or torque capabilities such that

$$-\tau_{i,\max} \leq \tau_i \leq \tau_{i,\max} \quad i = 1, \dots, N \quad (6)$$

a normalized force or torque can be defined as

$$\hat{\tau}_i = \frac{\tau_i}{\tau_{i,\max}} \quad (7)$$

and therefore, for the robotic system to be able to apply the desired wrench  $\mathbf{F}$ , the following condition must hold

$$\max_i |\hat{\tau}_i| \leq 1 \quad i = 1, \dots, N \quad (8)$$

Let us now define the point feasibility function  $U(\mathbf{p}_{EE}, \mathbf{F})$  of a point  $\mathbf{p}_{EE}$  in the kinematic workspace of the robotic system as

$$U(\mathbf{p}_{EE}, \mathbf{F}) = \max_i |\hat{\tau}_i| \quad i = 1, \dots, N \quad (9)$$

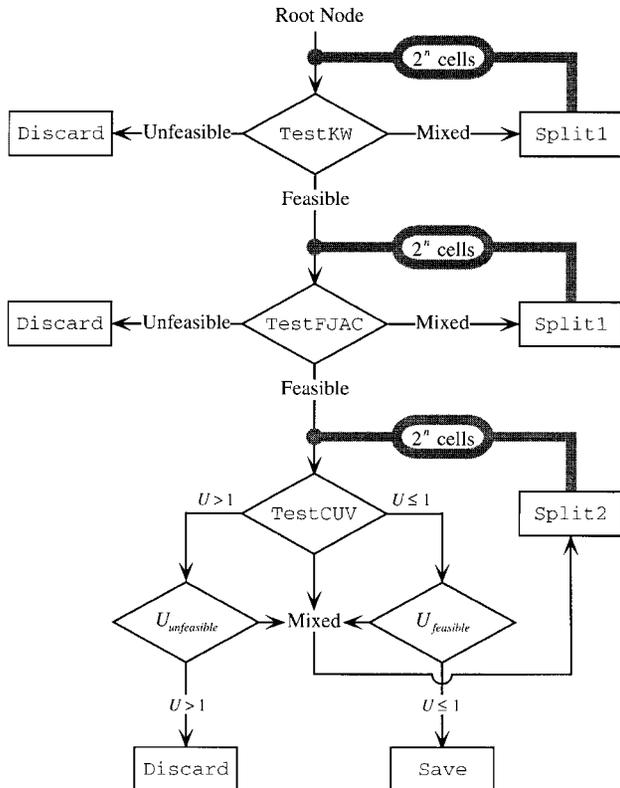


Figure 3. Force workspace generation flow chart.

where the  $\hat{\tau}_i$  represents the required actuator normalized force or torque such that the end-effector of the robotic system is applying a wrench  $\mathbf{F}$  at  $\mathbf{p}_{EE}$  in some configuration  $\mathbf{q}$ . Therefore, a point is feasible if the point feasibility function satisfies

$$U(\mathbf{p}_{EE}, \mathbf{F}) \leq 1 \quad (10)$$

To evaluate  $U(\mathbf{p}_{EE}, \mathbf{F})$ , the necessary actuator forces or torques  $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \cdots \ \tau_N]^T$  that will produce a desired end-effector wrench  $\mathbf{F}$  must be computed. For static or quasistatic operation of a serial manipulator, these can be expressed simply by

$$\boldsymbol{\tau} = \mathbf{G}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\mathbf{F} \quad (11)$$

where  $\mathbf{G}(\mathbf{q})$  is the vector of gravity terms and  $\mathbf{J}(\mathbf{q})$  is the Jacobian of the manipulator. In this work we assume that the number of actuated joints is equal to or exceeds the dimension of the desired wrench and therefore, that the robotic system can apply an arbitrary wrench. For any statically nonredundant manipulator, there exists a set of inverse kinematic functions  $\mathbf{f}_{IK}(\mathbf{p}_{EE})$ , with a finite number of solutions such that

$$\mathbf{q} = \mathbf{f}_{IK}(\mathbf{p}_{EE}) \quad (12)$$

We can therefore express the point feasibility function for a nonredundant robotic system as

$$U(\mathbf{p}_{EE}, \mathbf{F}) = \max_i |\hat{\tau}_i(\mathbf{f}_{IK}(\mathbf{p}_{EE}))| \quad i = 1, \dots, N \quad (13)$$

On the other hand, redundant manipulators typically have an infinite number of inverse kinematic solutions. Let us define  $\mathbf{q}_r$ , a subset of  $\mathbf{q}$ , as a set of redundant joint variables. We can obtain the configuration of a redundant manipulator using the following equation

$$\mathbf{q} = \mathbf{f}_{IK}(\mathbf{p}_{EE}, \mathbf{q}_r) \quad (14)$$

where  $\mathbf{f}_{IK}(\mathbf{p}_{EE}, \mathbf{q}_r)$  is a vector of inverse kinematic functions. To be part of the FW, points of the Cartesian space must correspond to end-effector locations where the end-effector can apply the desired wrench. Hence, for redundant manipulators, an end-effector location is part of the FW if there is at least *one* configuration for which this condition is true. This can be achieved by making sure that there exists some configuration at which the most loaded actuator does not exceed its limits. Mathematically, this

requires finding the maximum of the normalized actuator forces or torques and subsequently checking if the minimum value of this maximum, as the redundant joint variables take all possible values, is smaller than or equal to 1. Therefore, the point feasibility function for a redundant manipulator is defined as

$$U(\mathbf{p}_{EE}, \mathbf{F}) = \min_{\mathbf{q}_r} \max_i |\hat{\tau}_i(\mathbf{f}_{IK}(\mathbf{p}_{EE}, \mathbf{q}_r))| \quad i = 1, \dots, N \quad (15)$$

For a feasible workspace position, the condition given by Eq. (10) must still hold true.

Once the feasibility of a point is determined using the point feasibility criterion, Eqs. (10), and (13) or (15), we can proceed to determine the feasibility of a cell by setting up a *cell* feasibility criterion (CFC). For a cell to be entirely feasible (i.e., inside the FW), we require that all end-effector locations  $\mathbf{p}_{EE}$  in that cell have a point feasibility function value smaller than or equal to 1, i.e., that they all verify the PFC. If we define

$$U_{\text{feasible}} = \max_{\mathbf{p}_{EE} \in \text{cell}} U(\mathbf{p}_{EE}, \mathbf{F}) \quad (16)$$

then the cell feasibility criterion (CFC) is simply,

$$U_{\text{feasible}} \leq 1 \quad (17)$$

Similarly, for a cell to be *unfeasible* (i.e., outside the FW), we require that the PFC is never verified at all location in that cell. If we define

$$U_{\text{unfeasible}} = \min_{\mathbf{p}_{EE} \in \text{cell}} U(\mathbf{p}_{EE}, \mathbf{F}) \quad (18)$$

then the cell to be *unfeasible* is

$$U_{\text{unfeasible}} > 1 \quad (19)$$

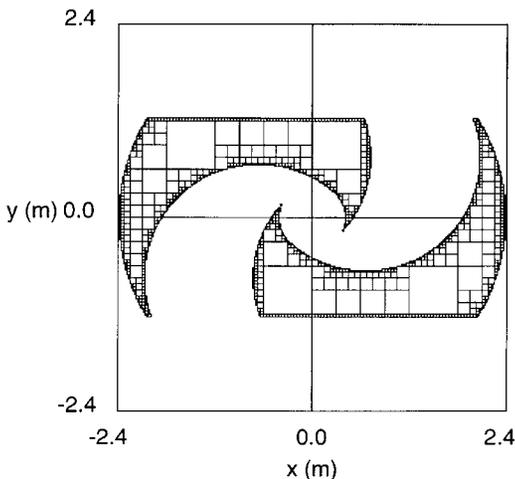
Whenever a cell has points for which the point feasibility function  $U$  takes values above and below 1, it is mixed, and has parts both inside and outside the FW. Since performing the computations dictated by Eqs. (16) or (18) is computationally intensive, we decided to check the  $U$  values at the vertices and the middle point of a cell, and subdivide the mixed cells right away (**TestCUV**).

However, this test is not sufficient to conclude on the overall feasibility of a cell; i.e., if it is located entirely inside or entirely outside the FW. To do this we must implement the CFC, hence Eqs. (17) or

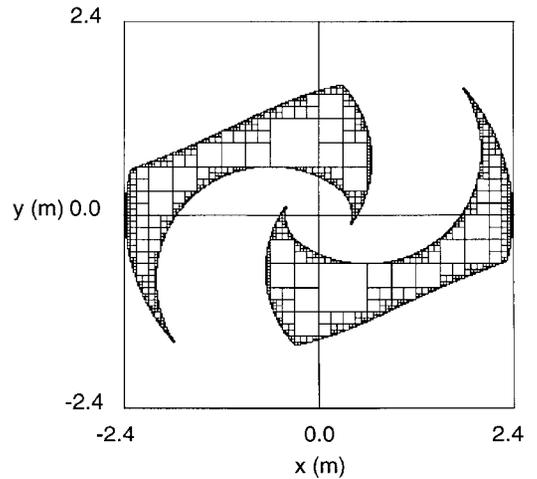
(19). If the  $U$  values found with **TestCUV** are all below 1, we use Eq. (17). We then verify that all points in that cell are feasible locations by computing  $U_{\text{feasible}}$ ; if  $U_{\text{feasible}}$  is greater than 1, then the cell is mixed and is further subdivided, otherwise it is feasible and is saved. If the  $U$  values found with **TestCUV** are all above 1, we use Eq. (19), and verify that all points in that cell are unfeasible locations by computing  $U_{\text{unfeasible}}$ . If  $U_{\text{unfeasible}}$  is smaller than 1, the cell is mixed and is further subdivided, otherwise it is unfeasible and is discarded. Cell subdivision stops at some predetermined resolution.

In the flow chart shown in Figure 3, the function **Split1** is used to subdivide the cells when no knowledge of the  $U$  function values at the vertices and middle point is required, while **Split2** is used when the  $U$  function value must be determined at the new vertices and middle points. The algorithm can be modified to find the intersection of force workspaces corresponding to a number of different force tasks. To minimize the amount of computations, the cells are deemed feasible at each test level only when they have been determined to be feasible for all tasks at that test level. For example, a cell is declared feasible by **TestFJAC** only if Eq. (5) is verified over the entire cell for all of the required tasks.

**Examples.** We generated a FW for the two-link planar manipulator used previously and depicted in Figure 1. The center of mass of each link is located at  $l_{CM_1} = 0.5$  m and  $l_{CM_2} = 0.3$  m and the mass of each link is  $m_1 = 0.30$  kg and  $m_2 = 0.25$  kg. The task was to apply a force of 8 N at  $0^\circ$ . Figure 4 shows the



**Figure 4.** Force workspace of a two-link manipulator without gravity.



**Figure 5.** Force workspace of a two-link manipulator with gravity.

result of the FW generation algorithm when gravity acts along the  $z$  axis, while Figure 5 shows the result when gravity acts along the negative  $t$  axis (i.e., the first joint actuator constraint approximation is not valid). Notice that gravity deforms the FW, assisting the force application at some points, and making it impossible at other points.

Using the FW generation algorithm described above, we also generated the feasible base locations for a redundant three-link SCARA-type manipulator for which gravity acts along the joint axes. The manipulator parameters are given in Table I. Here, two force tasks were prescribed. The first task,  $T_1$ , was to apply a force of 10.0 N at  $30^\circ$  at the position  $\mathbf{p}_{T_1} = [1.0 \ 1.0 \ 0.0]^T$  m, while the second task,  $T_2$ , was to apply a force of 10.0 N at  $-30^\circ$  at the position  $\mathbf{p}_{T_2} = [-1.0 \ 0.0 \ 0.0]^T$  m. Figure 6 depicts the computed locations at which the manipulator base must be placed so that both tasks can be accomplished without violating the actuator constraints of the manipulator. It is apparent that these limits have reduced the feasible base locations. Note that Figure 6 can also be used to place the base to a location that will tolerate greater force deviations with respect to the nominal desired force. Clearly, this can be achieved by placing the base as far away from the boundaries of the feasible base location area as possible.

#### IV. TASK WORKSPACE

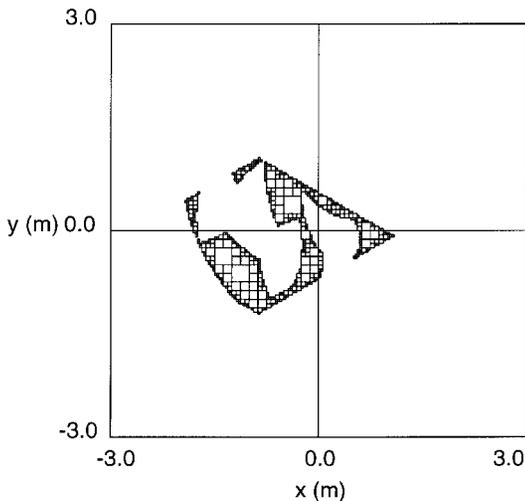
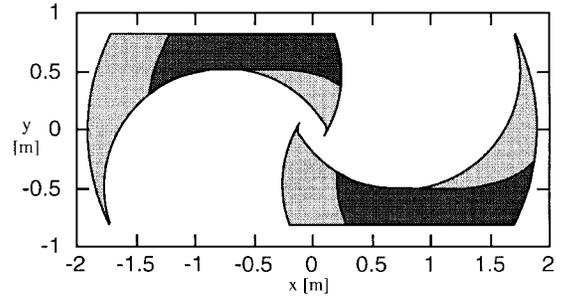
The above analysis focused on the application of some end-effector wrench at some particular loca-

**Table I.** Three-link manipulator parameters.

Link/Joint $i$	$l_i$ (m)	$\tau_{i,\max}$ (Nm)
1	1.4	10.0
2	1.0	5.0
3	0.6	3.0

tion. However, there are tasks for which wrenches must be applied along some given path. If the task is simple, such as applying a constant force in a constant direction, then the FW remains the same throughout the execution of the task. Therefore, the task is feasible if the path starting at some initial location is contained within the FW; i.e., placing the end-effector at the start location guarantees that the manipulator can accomplish the task without exceeding its actuator constraints. In general, a simple FW is not sufficient to determine if the manipulator can perform the entire task or not. We call the collection of all possible initial locations from which we can perform the task of applying some wrench along a given path the *task workspace* (TW). According to its definition, the TW is a subset of the FW.

To illustrate the concept of the task workspace, consider a two-link manipulator example with  $l_1 = 1.0$  m,  $l_2 = 0.9$  m, and  $\tau_{1,\max} = 10$  Nm,  $\tau_{2,\max} = 6$  Nm. The union of the light and dark gray areas in Figure 7 depicts the FW for applying a force of 12 N at  $0^\circ$ . In comparison, the dark gray region depicts the TW when the task is to apply the same force along a straight line of 0.5 m along the negative  $x$  direction.


**Figure 6.** Feasible base positions for two different tasks.

**Figure 7.** Comparison of the FW (light gray) and the TW for a two-link manipulator.

Note that the shape of the TW depends on the force-task, whose path may be arbitrary with an  $\mathbf{F}$  direction and magnitude changing along it. Therefore in general, the task workspace cannot be determined symbolically. However, a numerical algorithm such as the one proposed in this work can generate the TW or feasible base positions for an arbitrary task, even if no symbolic description of the task is available.

To obtain an accurate TW, each point of the initial force workspace must be validated for the entire path. The joint torque histories from the starting point and over the path must never exceed their respective actuator limits. Again, the quadtree (for 2D) or octree (for 3D) method can be implemented to decompose the workspace in a more efficient manner. The test function in this case must be validated for each starting point over the entire task before admitting it as part of the task workspace.

To generate the TW, the core of the algorithm used in the previous section is used again. The testing functions that determine the feasibility of a cell are modified to account for the fact that the wrench must now be applied along some arbitrary path. The kinematic feasibility test verifies that the path starting at any given cell location is contained at all times inside the kinematic workspace. Similarly, the first joint actuator constraint test verifies that the actuator torque limit of the first joint is never exceeded along the entire path.

For a path of length  $S$ , if  $s$  is the position of the end-effector along the path, then the normalized path length is defined as  $\hat{s} = s/S$ , and therefore  $0 \leq \hat{s} \leq 1$ . The test given by Eq. (5) is modified as

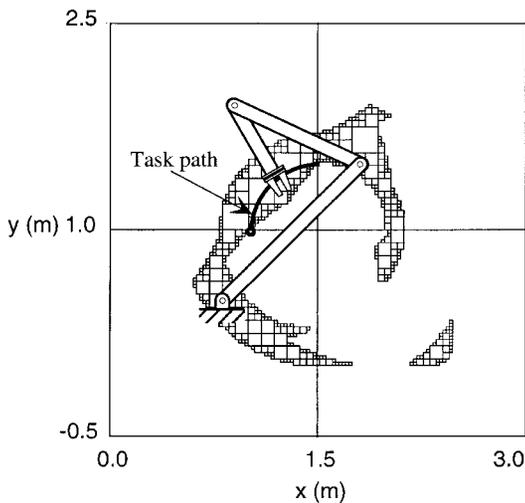
$$\max_s \left| (\mathbf{p}_{EE}(\mathbf{p}_{EE_0}, s) \times \mathbf{f}(s) + \mathbf{n}(s)) \cdot \hat{\mathbf{z}}_1 \right| \leq \tau_{1,\max} \quad (20)$$

where  $\mathbf{p}_{EE_0}$  refers to the starting location of the path. Also, the definition of the  $U$  function is modified to reflect the fact that the wrench is now being applied along a path,

$$U_s(\mathbf{p}_{EE_0}, \mathbf{F}) = \max_s \min_{\mathbf{q}_r} \max_i \left| \hat{\tau}_i(\mathbf{f}_{IK}(\mathbf{p}_{EE}(\mathbf{p}_{EE_0}, s), \mathbf{q}_r)) \right| \quad (21)$$

Replacing Eqs. (5) and (15) by Eqs. (20) and (21) allows us to use the FW generation algorithm for computing the TW.

**Example.** This modified algorithm was used to generate the TW of the same three-link planar manipulator used in the previous section. The task was to apply a constant force of 10 N directly toward the center of a quarter circle of radius 0.5 m. For this task, it is impossible to determine the shape of the TW from a simple inspection of the FW, as was done in Figure 7. Figure 8 shows the path of the task and the feasible base positions found using Eq. (2) when the starting position for the task is  $\mathbf{p}_T = [1.0 \ 1.0 \ 0.0]^T$  m. Also shown in the same figure is a mobile manipulator whose base has been placed to a position from which it is able to accomplish this task. This example also demonstrates the advantage of having manipulators with mobile bases in performing large force-tasks; for example, this manipulator would not have been able to execute



**Figure 8.** Feasible base positions for a three-link planar manipulator found using the TW.

this task if its base was located at the origin and was not mobile.

## V. CONFIGURATION PLANNING FOR REDUNDANT MANIPULATORS

### Optimal Configurations

A redundant manipulator has more degrees of freedom than a task requires, allowing one to choose additional conditions, which allow for redundancy resolution. Such conditions include energy minimization, joint limit avoidance, manipulability maximization, etc. A comprehensive review of geometric dynamic, or static criteria can be found in ref 11. To apply these, some manipulator performance index is defined first, and then optimized to result in planning configurations and trajectories. However, if these criteria are norm-based, they affect the manipulator in an overall manner; i.e., they cannot ensure that *individual* joint limitations are not violated. For example, optimizing the mechanical advantage, does not guarantee that individual joint actuator limits are not exceeded.

In the previous section, we referred to redundant manipulators and established a criterion for computing their force or task workspace. For a point to belong in those workspaces, there should be at least one configuration or one set of successive configurations at which the system can apply the desired wrenches.

In this section, we are interested in designing a force-task planner for redundant systems, able to select an optimal configuration or a succession of optimal configurations, so that the force capabilities are maximized. To do this, one must establish an appropriate criterion. This criterion should ensure necessarily that all individual joints are not required to exceed their force or torque limits. If configurations subject to this requirement do not exist, then the desired task is unfeasible. In addition, one may require that the normalized load of the most loaded actuator is minimized. This additional condition will then yield optimal manipulator configurations. Note that such a criterion has been defined already in Eq. (15), where the focus was identifying feasible points, and is repeated here in a slightly modified form,

$$\mathbf{q}_r^* \text{ such that } U = \min_{\mathbf{q}_r} \max_i |\hat{\tau}_i| \quad (22)$$

If the end-effector point of interest is in the FW of a system, then the  $U$  is less than or equal to 1.

Therefore, the first requirement discussed above is automatically met. Note however, that the criterion given by Eq. (22) allows one to find the optimal configuration  $\mathbf{q}^*$ , as defined above. This is because this criterion also provides the redundant joint angles  $\mathbf{q}_r^*$  at which the load of the most loaded joint is minimized. This optimal  $\mathbf{q}_r^*$ , along with the end-effector location and inverse kinematics, allows computing the optimal configuration  $\mathbf{q}^*$ . If the manipulator configuration is  $\mathbf{q}^*$ , then application of the given wrench will result in minimum loading of the most loaded actuator. All other actuators will be loaded less than this one, in a normalized sense.

### Force-Task Planning

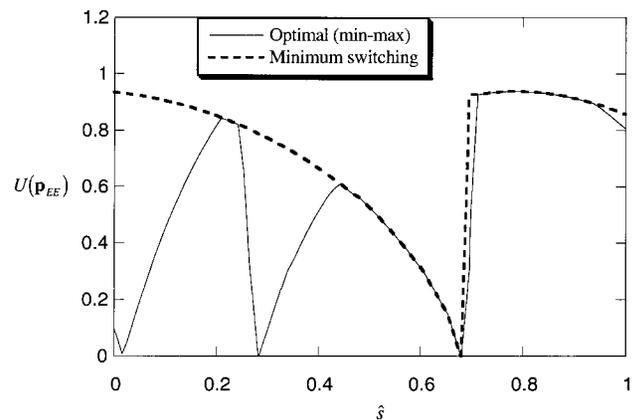
Force-task planning in redundant systems requires that a succession of optimal configurations for the manipulator is found as a function of the path end-effector. To solve this problem, a min-max optimization is performed as a function of the path parameter  $\hat{s}$ . The optimal  $\mathbf{q}^*$  is computed for the initial point on the path, where  $\hat{s} = 0$ , and the procedure is repeated by increasing  $\hat{s}$ , until the end of the path, where  $\hat{s} = 1$ . It can be shown that the resulting optimal configurations  $\mathbf{q}^*$  are piecewise continuous. However, there are path points at which the manipulator must change its posture while keeping its end-effector at the same location.<sup>14</sup> This can be explained as follows. Assume that actuator A is the most loaded one during motion along some path segment. At a switching point, actuator B becomes equally loaded to A, i.e., their normalized torques are equal. If for further motion of the end-effector along the path actuator B is the most loaded one, then Eq. (22) may result in a different set of optimal configurations, and therefore in a discontinuity in  $\mathbf{q}^*$ . Note that there is no need to perform the switching rapidly, as we are interested in path and not trajectory following. For example, removing an ORU with the special purpose dexterous manipulator (SPDM) requires application of a given wrench along a given path. If at some point along the path two of the manipulator joints are equally loaded and switching has to occur, then the end-effector stops moving until the manipulator slowly reconfigures itself. Once this happens, it will resume applying the desired wrench along the desired path. If during configuration switching, a manipulator's end-effector is not able to apply the desired wrench, then the particular combination of desired path and wrench is not feasible without repositioning the manipulator base.

In some cases, switching configurations may be undesirable or too time consuming. Then, one can reduce the switchings by selecting suboptimal configurations, i.e., ones for which actuator normalized forces or torques are below 1, but not necessarily minimum. However, in some cases switching cannot be avoided completely. Next, the min-max optimization along a path is illustrated using an example, and compared to a norm-based method.

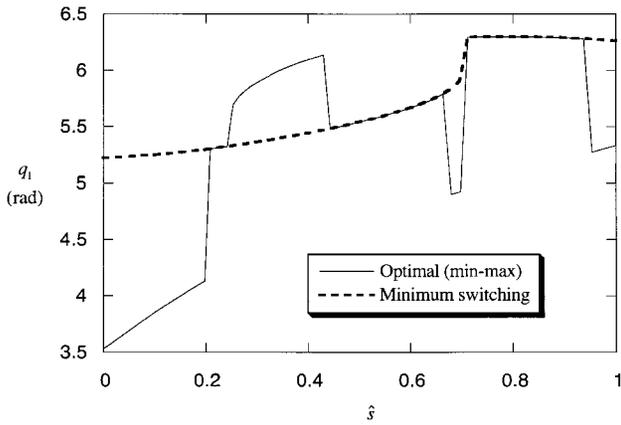
**Example.** The planning method described above is applied here to the three-link SCARA manipulator whose parameters are given in Table I. The force-task consists of applying a force of 8 N at  $0^\circ$  along a straight line connecting points A and B with coordinates  $(x_A, y_A) = (0.3, -0.6)$  m and  $(x_B, y_B) = (2.3, -0.6)$  m, i.e., a motion parallel to the  $x$  axis. For this system, Eq. (22) becomes

$$q_1^* \text{ such that } \min_{q_1} \max_i |\hat{\tau}_i(\mathbf{f}_{IK,3L}(\mathbf{p}_{EE}, q_1))| \quad i = 2, 3 \quad (23)$$

where  $\mathbf{f}_{IK,3L}(\mathbf{p}_{EE}, q_1)$  is the vector of inverse kinematic functions for a three-link planar manipulator when joint angle  $q_1$  is chosen as the redundant joint variable. However, any other angle could have been used. Figure 9 depicts the result of the min-max optimization in the form of the maximum normalized torque,  $U(\mathbf{p}_{EE}, \mathbf{F})$ , as a function of the path parameter  $\hat{s}$ . Figure 10 displays the corresponding solution for  $q_1^*$ . Discontinuities in this figure correspond to switchings from one configuration to another. To minimize the number of switchings, one



**Figure 9.** Maximum normalized torque vs. path parameter  $\hat{s}$  corresponding to the optimal and minimum switching solutions.



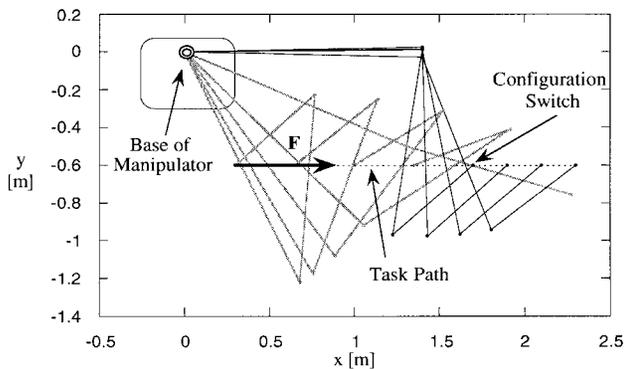
**Figure 10.** Optimal  $q_1$ -configurations along path  $\hat{s}$ .

can use a suboptimal solution as shown in Figure 9. This suboptimal solution requires a single switching at  $s = 0.7$ . Figure 10 displays the resulting solution for  $q_1$ , while Figure 11 displays the evolution of configurations along the path when the minimum switching solution is used.

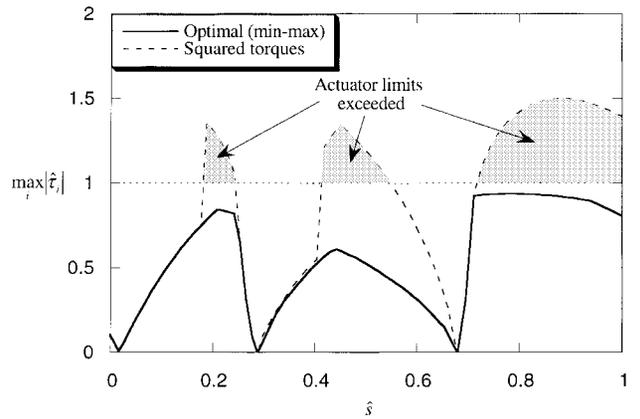
Next, the proposed method is compared to a frequently used method which resolves the redundancy by minimizing the sum of weighted actuator squared torques as follows,

$$q_1^* \text{ satisfies } \min\{w_1\tau_1^2(q_1) + w_2\tau_2^2(q_1) + w_3\tau_3^2(q_1)\} \quad (24)$$

Weights  $w_i$  ( $i = 1, 2, 3$ ) are needed especially in the case where prismatic and rotary actuators coexist so



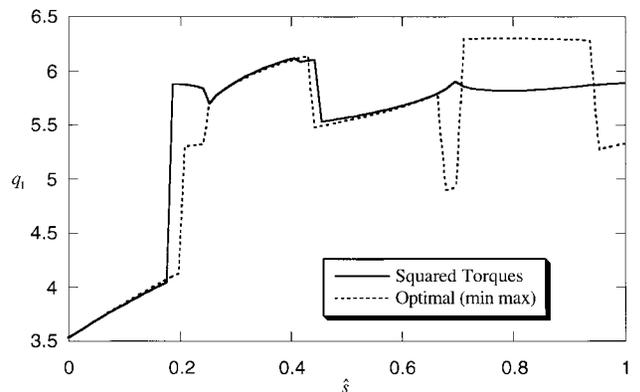
**Figure 11.** Snapshots of configurations for the minimum switching solution.



**Figure 12.** Maximum normalized torques comparison for the min-max optimal and the squared-torques methods.

that the criterion is dimensionally homogeneous. Note that the min-max method proposed here does not suffer from this limitation. Also note that the criterion given by Eq. (24) does not guarantee that during a force-task no actuator will saturate. Instead, it guarantees that the “power” required is minimum.

To compare the solution obtained previously by using the min-max criterion, to the one resulting from the use of Eq. (24), we apply the latter using unit weights. In such a case, Eq. (24) roughly minimizes total power consumption. The normalized torques computed using the min-max optimal and the squared-torque method are shown in Figure 12, while the corresponding  $q_1$ -configurations are shown in Figure 13. Figure 12 reveals that the maximum normalized torque computed by the



**Figure 13.** Joint angle  $q_1$  comparison for the min-max optimal and the squared-torques methods.

squared-torque method exceeds the maximum available, 1.0, for a substantial part of the task. Therefore, in the case where the desired end-effector force is large, such a method fails to yield feasible posture planning. On the other hand, the min-max optimal method yields normalized torques that are always below one. It should be noted that changing the weights in Eq. (24) does not alleviate the problem; for example, increasing the weight on joint 3 does result in lower torques for it, but then the other joint actuators do saturate.

## VI. CONCLUSIONS

This article presented a framework for the problem of applying large wrenches with robotic systems of limited force or torque actuator capabilities. It was shown that the configurations in which such systems can apply a large wrench are limited. Mapping these configurations to the Cartesian space gives rise to the concept of the force workspace (FW). An efficient numerical algorithm based on the  $2^n$ -tree decomposition of the Cartesian space was proposed to generate the FW of a manipulator. It was shown that force capabilities can be improved by employing base mobility and manipulator redundancy. Based on the FW generation algorithm, a planning method was presented which results in proper base positioning relative to large-force quasistatic tasks. In addition, the case of tasks requiring the application of a wrench along a given path was considered. The task workspace, the set of Cartesian space locations that are feasible starting positions for such tasks, was shown to be a subset of the force workspace. This workspace can be used for identifying proper base or task positions that will guarantee task execution along desired paths. Finally, to plan redundant manipulator postures during large force-tasks, a new method based on a min-max optimization scheme was developed. Unlike other norm-based methods, this method guarantees that no actuator capabilities are exceeded, and that the force or torque of the most loaded joint is minimized. Illustrative examples were given that demonstrate the validity and usefulness of the proposed framework.

The support of this work by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR), is gratefully acknowledged.

## REFERENCES

1. J.E. Bobrow, S. Dubowsky, and J.S. Gibson, Time-optimal control of robotic manipulators along specified paths, *Int J Robot Res* 4 (1985), 3–17.
2. J. Hollerbach and K.C. Suh, Redundancy resolution of manipulators through torque optimization, *IEEE Trans Robot Automat* 3 (1987), 308–316.
3. D.E. Orin and S.Y. Oh, Control of force distribution in robotic mechanisms containing closed kinematic chains, *ASME J Dyn Syst Meas Contr* 102 (1981), 134–140.
4. Y. Nakamura, Force applicability of robotic mechanisms, *Proc 26th Conference Decision and Control*, 1987, pp. 570–575.
5. S. Chiu, Control of redundant manipulators for task compatibility, *Proc IEEE Int Conf Robotics & Automation*, 1987.
6. Z. Li, T.J. Tarn, and A. Bejczy, Dynamic workspace of multiple cooperating robot arms, *IEEE Trans Robot Automat*, 7 (1991), 589–596.
7. J. Lenarcic and L. Zlajpah, Comparison of local and global solution in optimisation of joint torques of  $n$ -R planar manipulator, *Proc IFAC Symposium Robot Control (SYROCO)*, 1994, pp. 447–452.
8. B. Nemeč, Control of redundant manipulators with limited torque, *Proc IFAC Symposium on Robot Control (SYROCO)*, 1994, pp. 461–466.
9. B.J. Martin and J.E. Bobrow, Determination of minimum-effort motions for general open chains, *Proc IEEE Int Conf Robotics & Automation*, 1995, pp. 1160–1165.
10. A. Madhani and S. Dubowsky, Motion planning of mobile multi-limb robotic systems subject to force and friction constraints, *Proc IEEE Int Conf Robotics & Automation*, 1992, pp. 233–239.
11. H. Seraji, Task-based configuration control of redundant manipulators, *J Robot Syst* 9 (1992), 411–451.
12. B. Paden, A. Mees, and M. Fisher, Path planning using Jacobian freespace generation, *Proc IEEE Int Conf Robotics & Automation*, 1989, pp. 1732–1737.
13. H. Samet, *The design and analysis of spatial data structures*, Addison-Wesley, Reading, MA, 1990.
14. Y. Gonthier, Force task planning of robotic systems with limited actuator capabilities, M.S. Thesis, Dept. of Mechanical Engineering, McGill University, Montreal, Canada, 1996.