Robotica (2002) volume 20, pp. 625–636. © 2002 Cambridge University Press
DOI: 10.1017/S0263574702004204 Printed in the United Kingdom

On-line multi-criteria based collision-free posture generation of redundant manipulator in constrained workspace

Jin-Liang Chen, Jing-Sin Liu,* Wan-Chi Lee and Tzu-Chen Liang

Institute of Information Science 20, Academia Sinica, Nankang, Taipei 115, Taiwan (R.O.C.)

(Received in Final Form: March 22, 2002)

SUMMARY

The manipulator with a large degree of redundancy is useful for realizing multiple tasks such as maneuvering the robotic arms in the constrained workspace, e.g. the task of maneuvering the end-effector of the manipulator along a pre-specified path into a window. This paper presents an on-line technique based on a posture generation rule to compute a null-space joint velocity vector in a singularity-robust redundancy resolution method. This rule suggests that the end of each link has to track an implicit trajectory that is indirectly resulted from the constraint imposed on tracking motion of the end-effector. A proper posture can be determined by sequentially optimizing an objective function integrating multiple criteria of the orientation of each link from the end-effector toward the base link as the secondary task for redundancy resolution, by assuming one end of the link is clamped. The criteria flexibly incorporate obstacle avoidance, joint limits, preference of posture in tracking, and connection of posture to realize a compromise between the primary and secondary tasks. Furthermore, computational demanding of the posture is reduced due to the sequential link-by-link computation feature. Simulations show the effectiveness and flexibility of the proposed method in generating proper postures for the collision avoidance and the joint limits as a singularity-robust null-space projection vector in maneuvering redundant robots within constrained workspaces.

KEYWORDS: Posture generation; Redundant manipulator; Constrained workplace; Collision avoidance.

1. INTRODUCTION

A robotic manipulator is called kinematically redundant if it possesses more degrees of freedom (DOF) than is necessary for performing a specified task at the end-effector. The number of DOF is determined by the kinematic structure of a manipulator, which usually coincides with the number of independently controlled drives (i.e. joints). The extra DOF provides an infinite number of joint motions corresponding to the same configuration of the end-effector. This feature increases the dexterity and the versatility of the robot in performing tasks. Significantly, the redundancy in a manipulator structure has been applied for avoiding obstacles[1–19] joint limits, and singularity of kinematic matrices,[20,21] or peak torque reduction,[22] torque optimization,[23] and joint failure/fault tolerance.[24]

Collision avoidance is a basic capability for the redundant manipulator, which ensures the robot links to avoid obstacles cluttered in the environment while performing the primary tasks at the end-effector. A proper posture of redundant robot needs to be generated for use in obstacle avoidance tracking control[25] or collision-free motion planning[2] as a desired joint trajectory. In practice, obstacle avoidance can be solved by either off-line[1–6,26–28] or on-line planning. The high-level path planning globally finds a collision-free path in the configuration space (i.e. joint space of the manipulator) before performing tasks.[26–28] However, searching in high-dimensional configuration space of the redundant manipulator is actually not efficient. Therefore, some methods tried to directly solve the problem in workspace by off-line planning collision-free configurations of the robot at many successive points of a given path of end-effector,[1,2] or by heuristically and frequently adjust the links of the robot away from collision with obstacles[3–5] for planning collision-free trajectory. Furthermore, McLean and Cameron[6] proposed a virtual spring method for quick path planning in the workspace that is modeled by the artificial potential field and useful for avoiding local minima in the potential field.

Alternatively, obstacle avoidance can be solved on-line by the robot controller at the low level.[7–19] A redundant robot is controlled so that the end-effector tracks a given path in the workspace as closely as possible and simultaneously ensures that the links avoid obstacles. Glass *et al.*[7] proposed an approach that represents the requirement of collision avoidance by a set of kinematic inequality constraints for implementing configuration control,[29] and then ensures that active constraints are satisfied in tracking the given path. In Reference 8, Newman proposed a concept of reflex control so that the control command given from a high-level planner to the controller could be interfered when the robot may collide with obstacles. However, collision avoidance using the reflex control scheme depends on the search of configuration space (that is high-dimensional) for guiding the interference right enough.[9] Based on the robot dynamic control in the operational space, Khatib[10] proposed a method using the artificial potential field in the operational space to generate joint torques by the force of potential gradient.

Besides, the null-space projection of primary task Jacobian pseudoinverse is a well-known technique for controlling kinematically redundant manipulators.[20] Gen-

* Corresponding author: E-mail: liu@iis.sinica.edu.tw

erally, works performed at the endeffector is the primary task and utilization of the redundancy refers to the secondary (or constraint) tasks. The technique of task-priority redundancy resolution[11] has the advantage that the constraint tasks can be executed through the null-space projection of pseudoinverse without affecting the primary task. Thus, the technique is usually employed to optimize an objective function for obstacle avoidance.[12–19] Maciejewski and Klein[17] proposed a method based on the task-priority technique[11] that dynamically identifies the point on the link of manipulator that is closest to an obstacle and assigns to it a desired velocity vector away from the obstacle surface. The method is applied to obstacle avoidance in workspace where only one link is constrained by obstacles. However, the velocities assigned to these obstacles avoidance points have to be specified heuristically. This may result in oscillations of robot's links in narrow paths when magnitudes of the velocities are specified too large. In reference 18, a velocity potential field, which has no local minima while all obstacles are convex and there exist gaps between the obstacles, is utilized to specify the velocities away from obstacles. Choi and Kim[12,13] represented obstacles and links of the robot as spheres and ellipsoids, respectively. This method predicts collision between the sphere and the ellipsoid via measuring their directional and temporal meeting, and then assigns escaped velocities to joints based on gradient of analytic function of the collision measure. Similarly, the method in reference 14 has to assume that the distance between each obstacle and each link is a differentiable function, i.e., an analytic equation. Besides, the contributions of Rahmanian-Shahri and Troch[15,16] tried to specify one or several ellipse(s) to enclose an obstacle as the barrier limiting a particular joint to be located in the workspace, and keep each of joints out of its corresponding barriers.

In our previous work,[2] a combination of potential field and elastic model method is used to generate a proper posture of redundant robot for off-line collision avoidance in constrained workspace. Multiple links may be navigated into the workspace and thus their movements are constrained by the obstacles. Assigning to the closest link an obstacle avoidance vector may result in other links to collide with obstacles. The movement of each link for successful collision avoidance is interactive due to the constraints imposed by the obstacles. This interaction is revealed by a posture generation rule presented in this paper, and is useful for on-line planning posture of the redundant manipulator with larger degrees of freedom. Then, an optimization technique is proposed for implementing this rule. The rule suggests that the end of each link has to track an implicit path. The implicit path for each link is indirectly resulted from the constraint motion specified for the endeffector. By requiring the end of each link to follow an implicit path, only link orientation can be adjusted for the constrained tasks, e.g. avoidance of collision or joint limit. Therefore, the link orientations could be determined sequentially by a particular sequence (backwards from end-effector to base link). Based on this rule, multiple objective functions are developed for determining the link orientations. By following the gradient of a combination of these functions, the rotation velocity of each link orientation, which has a tendency to maximize the combined objective function, can be obtained. By this way, a set of collision-free nullspace joint velocities can be obtained link-by-link. Furthermore, by the use of the task-priority redundancy resolution, a proper posture of the manipulator is approached without affecting the motion of the end-effector.

This paper is organized as follows First, task-priority redundancy resolution by singularity-robust method is briefly reviewed in Section 2. The posture generation rule for sequentially orienting links is presented in Section 3, and the corresponding multi-criterion based technique is proposed in Section 4. Although the development is mainly for planar redundant robots, simulations results in Section 5 show the performance of the proposed approach for planar as well as spatial redundant robots for obtaining a null-space joint velocity vector for tackling simultaneously obstacle avoidance and joint limits. Finally, conclusions are made in Section 6.

2. TASK-PRIORITY REDUNDANCY RESOLUTION

2.1. Inverse Kinematics

Consider a redundant manipulator with n degrees of freedom in m dimensional workspace, where $n>m$. And, the end-effector of the robot has to track a given path $\mathbf{x}_E \in R^m$, which is a function of time t may be specified in advance or on-line determined by a joystick. The forward kinematics mapping the joint vector $\boldsymbol{\theta} \in R^n$ to $\mathbf{x}_E$ can be represented as the function

$$\mathbf{x}_E = f(\boldsymbol{\theta}). \tag{1}$$

The forward rate kinematics is

$$\dot{\mathbf{x}}_E = \mathbf{J}_E \dot{\boldsymbol{\theta}} \quad \text{and} \quad \mathbf{J}_E = df(\boldsymbol{\theta})/d\boldsymbol{\theta}, \tag{2}$$

where $\mathbf{J}_E \in R^{m\times n}$ is a *Jacobian* matrix of the end-effector.[20] For obtaining the joint rate velocity $\dot{\boldsymbol{\theta}}$, the general solution of (2) is

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_E^+ \dot{\mathbf{x}}_E + (\mathbf{I} - \mathbf{J}_E^+ \mathbf{J}_E)\mathbf{z}, \tag{3}$$

where $\mathbf{J}_E^+$ denotes the pseudoinverse of $\mathbf{J}_E$, $\mathbf{I} \in R^{n\times n}$ is the identity matrix, and $\mathbf{z}$ is an arbitrary vector in the joint velocity space which determines how to use the robotic redundancy. The pseudoinverse $\mathbf{J}_E^+$ could be computed based on the undetermined case ($m<n$) or overdetermined case ($n<m$), for which $\mathbf{J}_E^+ = \mathbf{J}_E^T(\mathbf{J}_E\mathbf{J}_E^T)^{-1}$ or $\mathbf{J}_E^+ = (\mathbf{J}_E^T\mathbf{J}_E)^{-1}\mathbf{J}_E^T$. It is well known that $\mathbf{J}_E^+ \dot{\mathbf{x}}_E$ is the minimum-norm solution of $\|\dot{\boldsymbol{\theta}}\|$ and $(\mathbf{I} - \mathbf{J}_E^+ \mathbf{J}_E)$ is the null-space projection matrix.[20] The homogeneous term $(\mathbf{I} - \mathbf{J}_E^+ \mathbf{J}_E)\mathbf{z}$ is orthogonal to $\mathbf{J}_E^+ \dot{\mathbf{x}}_E$, i.e., $(\mathbf{I} - \mathbf{J}_E^+ \mathbf{J}_E)^T \mathbf{J}_E^+ = \mathbf{0}$, which results in self-motion of the manipulator without affecting the motion $\dot{\mathbf{x}}_E$ of the end-effector, i.e., $\mathbf{J}_E(\mathbf{I} - \mathbf{J}_E^+ \mathbf{J}_E)\mathbf{z} = 0$. In other words, the secondary tasks of joint limits and obstacle avoidances can be realized by the use of null-space projection.

2.2. Task-priority Redundancy Resolution

Nakamura and Yoshikawa[11] proposed the task-priority redundancy resolution scheme for performing the secondary tasks (i.e. the constraint tasks) without affecting the primary

task. Similar to (1) and (2), the constraint task is defined as the forward rate kinematics

$$\dot{\mathbf{x}}_C = \mathbf{J}_C \dot{\boldsymbol{\theta}} \tag{4}$$

where $\dot{\mathbf{x}}_C$ and $\mathbf{J}_C$ denote the specification of constraint task and the corresponding Jacobian matrix, respectively. To solve (3) to meet (4), substituting (3) for $\dot{\boldsymbol{\theta}}$ into (4) and then solving the free vector $\mathbf{z}$, we have

$$\mathbf{z} = \tilde{\mathbf{J}}_C^{+}(\dot{\mathbf{x}}_C - \mathbf{J}_C \mathbf{J}_E^{+} \dot{\mathbf{x}}_E) + (\mathbf{I} - \tilde{\mathbf{J}}_C^{+} \tilde{\mathbf{J}}_C)\mathbf{y}, \tag{5}$$

where $\tilde{\mathbf{J}}_C = \mathbf{J}_C(\mathbf{I} - \mathbf{J}_E^{+} \mathbf{J}_E)$ and $\mathbf{y}$ denotes another arbitrary vector in the joint velocity space. The matrix $\tilde{\mathbf{J}}_C$ gives the available range for the secondary task to perform without affecting the primary task. Let $\mathbf{y} = \mathbf{0}$ and substitute (5) back to (3), the desired solution for $\dot{\boldsymbol{\theta}}$ is

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_E^{+} \dot{\mathbf{x}}_E + (\mathbf{I} - \mathbf{J}_E^{+} \mathbf{J}_E)\tilde{\mathbf{J}}_C^{+}(\dot{\mathbf{x}}_C - \mathbf{J}_C \mathbf{J}_E^{+} \dot{\mathbf{x}}_E). \tag{6}$$

Since the projection operator $(\mathbf{I} - \mathbf{J}_E^{+} \mathbf{J}_E)$ is both Hermitian and idempotent, (6) can be simplified to

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_E^{+} \dot{\mathbf{x}}_E + \tilde{\mathbf{J}}_C^{+}(\dot{\mathbf{x}}_C - \mathbf{J}_C \mathbf{J}_E^{+} \dot{\mathbf{x}}_E). \tag{7}$$

However, as the controller (3) or (7) is applied, the algorithmic singularity problem, i.e. high joint velocities and oscillations, occurs while the matrix $\mathbf{J}_E$ or $\tilde{\mathbf{J}}_C$ loses rank.

2.3. Singularity-Robust Resolution

The problem of matrix singularity results in outputs of very large joint velocities that are usually unacceptable in practical implementation. This problem can arise from two situations.

(i) the kinematic singularities happen when either the kinematic matrix $\mathbf{J}_E$ or $\mathbf{J}_C$ is not of full rank.
(ii) the task-priority strategy leads to the algorithmic singularities occurred at which the matrix $\tilde{\mathbf{J}}_C$ loses rank with full rank $\mathbf{J}_C$ and $\mathbf{J}_E$.

In order to handle both kinematic and algorithmic singularities, a singularity-robust task-priority redundancy resolution based on the damped least-squares method[30] is proposed by Chiaverini.[21] For the kinematic singularities, the damped least-squares inverse $\mathbf{J}^{+}$ would be used to replace the original pseudoinverse $\mathbf{J}^{+}$. It is obtained by incorporating a damping factor λ (a small positive number, e.g. $\lambda = 0.01$) into computing the pseudoinverse[30]

$$\mathbf{J}^{*} = \mathbf{J}^{T}(\mathbf{J}\,\mathbf{J}^{T} + \lambda^{2}\mathbf{I})^{-1}. \tag{8}$$

On the other hand, Chiaverini[21] shows that the algorithmic singularities of $\tilde{\mathbf{J}}_C$ can be decoupled from the singularities of $\mathbf{J}_C$, and then revise (6) by the following equation

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_E^{*} \dot{\mathbf{x}}_E + (\mathbf{I} - \mathbf{J}_E^{*} \mathbf{J}_E)\mathbf{J}_C^{*} \dot{\mathbf{x}}_C. \tag{9}$$

One should be noted that the joint velocity associated with the constrained task is projected onto the nullspace of $\mathbf{J}_E$ to remove the components that would interfere with the end-effector (i.e. primary) tracking task.

In Section 3 and Section 4, we present a posture generation rule, mainly for planar redundant robots, and a multi-criteria gradient projection technique to calculate a null-space joint velocity vector as $\mathbf{J}_C^{*} \dot{\mathbf{x}}_C$ in (9) for accomplishing the secondary tasks, involving the obstacles and joint limit avoidances.

3. POSTURE GENERATION RULE

3.1. Specification of Redundant Manipulator

In the paper, we study planar or 3D redundant robots with revolute joints. For a planar manipulator, two subsequent links are connected by a joint. For example, Figure la shows a planar manipulator in the *zy*-plane and each of the joints having the rotation axis parallel to the *x*-axis. That is, orientation of each link vector is determined by the rotation of a revolute joint. In general, forward kinematics of each *link vector* could be represented by

$$\vec{O}_i = \mathbf{R}(\theta_i) \cdot \ldots \cdot \mathbf{R}(\theta_1) \cdot -\vec{L}_i, \tag{10}$$

where $\vec{O}_i \in R^2$ denotes the vector linking the joints $\mathbf{x}_{i+1}$ and $\mathbf{x}_i$, i.e. $\vec{O}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, $\mathbf{R}(\theta_j) \in R^{2\times 2}$ denotes the rotation matrix of θ_j, and $\vec{L}_i \in R^2$ denotes the link vector when $\theta_j = 0$ for $j = 1$ to i.

Remark. On the other hand, consider a 3D redundant robot with two links connected by a pair of revolute joint whose rotation axes are perpendicular to each other, which is in order to simolate a spherical joint with two degrees of freedom. For example, Figure 1b shows the structure of a 3-D redundant manipulator in which two subsequent links are assumed connected by a pair of joints (θ_i^x, θ_i^z) that rotation axes are parallel to the *x*- and *z*-axes, respectively. Such pair of joints describes a spherical coordinate with consistent length of the link vector, which can pose each link vector with arbitrary direction. The kinematics of each link vector is represented by

$$\vec{O}_i = \mathbf{R}(\theta_i^x, \theta_i^z) \cdot \ldots \cdot \mathbf{R}(\theta_1^x, \theta_1^z) \cdot \vec{L}_i, \tag{11}$$

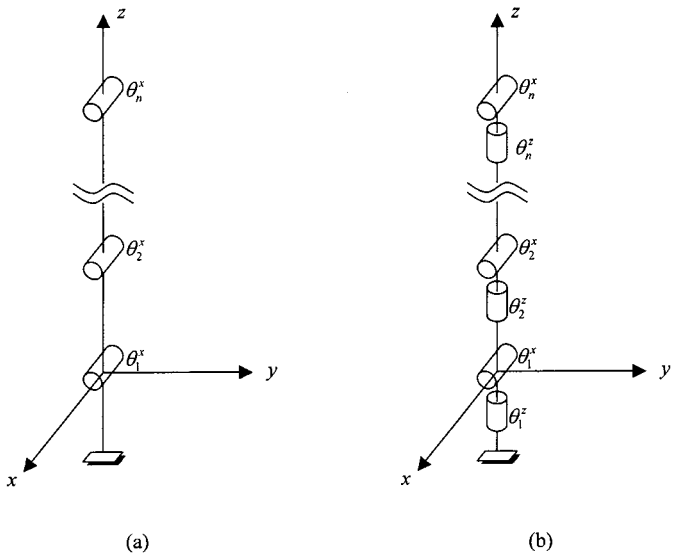

Fig. 1. (a) A planar manipulator in the *zy*-plane and each of the joints having the rotation axis parallel to the *x*-axis. (b) a 3-D redundant manipulator in which two links are connected by a pair of joints (θ_i^x, θ_i^z) that rotation axes are parallel to the *x*- and *z*-axes, respectively.

where $\vec{O}_i$, $\vec{L}_i \in R^3$, and

$$\mathbf{R}(\theta_i^x, \theta_i^z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i^x & \sin\theta_i^x \\ 0 & -\sin\theta_i^x & \cos\theta_i^x \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_i^z & -\sin\theta_i^z & 0 \\ \sin\theta_i^z & \cos\theta_i^z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

Based on the robotic kinematics described above, the posture generation rule is introduced as follows.

3.2. Posture Generation Rule

The posture generation rule proposed here is observed from the sequential motion of each joint deduced from the end-effector motion. For the convenience of presentation, consider a planar redundant manipulator with n DOFs. Figure 2 depicts a manipulator in which the end-effector attached to the n^{th} link move with a desired velocity vector $\dot{\mathbf{x}}_E$. Due to the constraint of $\dot{\mathbf{x}}_E$ at the end-effector, avoidance of collision on the n^{th} link inevitably depends on its orientation. Let $\vec{r}_n$ denote the displacement at the n^{th} joint as the orientation of the n^{th} link is changed and the end $\mathbf{x}_E$ is assumed clamped due to the constraint at the end of the link. Then the $(n-1)^{\text{th}}$ link follows an implicit path $\dot{\mathbf{x}}_n = \dot{\mathbf{x}}_E + \vec{r}_n$ at its end (i.e. the n^{th} joint). By induction, for i^{th} link

$$\dot{\mathbf{x}}_i = \dot{\mathbf{x}}_{i+1} + \vec{r}_i$$

where $\dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_E$ is the pre-specified end-effector motion. That is, the posture of each link is determined by a translation (i.e. $\dot{\mathbf{x}}_{i+1}$) at one end and a rotation (i.e. $\vec{r}_i$) at the other end. Then the process goes on to next link (link $i-1$). The above process is iterated to determine a desired motion of links until the base links (1st link). In general, the implicit path can be written as the backwards recursive form

$$\dot{\mathbf{x}}_i = \dot{\mathbf{x}}_{i+1} + \vec{r}_i, \; i = n, \ldots, 0 \quad (13)$$

$$\vec{r}_i(\dot{\theta}_i) = J_c \cdot \dot{\theta}_i \quad \text{and} \quad J_c = d(-\vec{O}_i)/d\theta_i$$

with two boundary constraints

$$\dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_E \text{ given end-effector velocity}$$

$$\dot{\mathbf{x}}_0 = 0 \text{ for immobile base}$$

A posture determination rule deduced from (13) is proposed as follows. Posture of the redundant manipulator can be planned by sequentially optimizing the orientations of links from the end-effector toward the base. Each link orientation (or $\vec{r}_i$) is planned by assuming the end of the link (or $\dot{\mathbf{x}}_{i+1}$) is clamped, owing to the constraint of implicit path at the end of the link. For a collision-free configuration, there is an

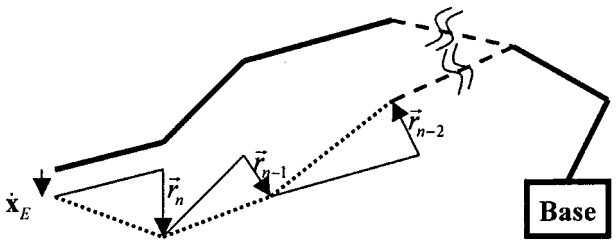

Fig. 2. An illustration of the posture generation rule: the motion of each joint $\dot{\mathbf{x}}_i = \dot{\mathbf{x}}_{i+1} + \vec{r}_i$ is induced by end-effector $\dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_E$.

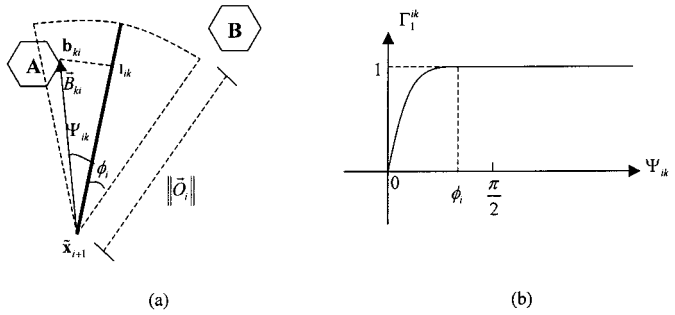

Fig. 3. (a) the dashed orientation cone indicates the allowable range that the link vector $-\vec{O}_i$ can be moved if any obstacle intersects this cone. (b) the plot of the objective function Γ_1^{ik} versus the included angle Ψ_{ik} between $-\vec{O}_i$ and $-\vec{B}_{ki}$.

allowable collision-free moving range of the link with one-end clamped. This free range of joint rotation spans a cone in workspace. The size of this cone depends on the distance between the link and the obstacles. An illustration is depicted in Figure 3a, in which $\dot{\mathbf{x}}_{i+1}$ is assumed fixed, the convex polygons indicate obstacles cluttered around the link, and the dashed cone denotes the allowable range of link orientations for avoiding collisions. Other parameters shown in Figure 3a are introduced later in Section 4. On the other hand, the displacement $\vec{r}_i$ would result in a translation $\dot{\mathbf{x}}_i$ (by (13)) of its adjacent $(i-1)^{\text{th}}$ link. Due to such interacting effect, collision avoidance of the adjacent link should be taken into account when $\vec{r}_i$ is optimized. Besides, the other criteria, e.g., respecting the joint limits, can also be incorporated in the optimization of $\vec{r}_i$. This is addressed in Section 4.

By employing this rule to determine postures, an obvious advantage is that the computational load of on-line posture planning can be reduced, because the proposed method for determining a posture is link-by-link computed, starting from the end-effector to the base, and requires one calculation of $\vec{r}_i$ for each link. Based on the use of the singularity-robust resolution (9), the constraint task can be specified by

$$\dot{\mathbf{x}}_C = -[\vec{r}_1^T \cdots \vec{r}_n^T]^T, \quad (14)$$

where $\dot{\mathbf{x}}_C \in R^{(n\times m)\times 1}$. By the rule, each $\vec{r}_i$ defines the optimized change of the link orientation. Therefore, the kinematic matrix $\mathbf{J}_C$ for $\dot{\mathbf{x}}_C$ would be computed by

$$\mathbf{J}_C = \begin{bmatrix} \frac{d\Omega}{d\theta_1} & \cdots & \frac{d\Omega}{d\theta_n} \end{bmatrix}^T, \quad \text{where } \Omega = -[\vec{O}_1^T \quad \cdots \quad \vec{O}_n^T]^T, \quad (15)$$

where $\mathbf{J}_C \in R^{(n\times m)\times n}$. Then, the singularity-robust pseudo-inverse of $\mathbf{J}_C$ is obtained as

$$\mathbf{J}_C^* = (\mathbf{J}_C^T \mathbf{J}_C + \lambda^2 \mathbf{I})^{-1} \mathbf{J}_C^T. \quad (16)$$

4. MULTI-CRITERIA FOR SEQUENTIAL DETERMINATION OF LINK ORIENTATIONS

Accordingly, the posture generation rule is suitably applied to any redundant manipulator with larger degrees of freedom. In this section, a multi-criteria based technique is

developed to compute the orientation of each link sequentially. A combination of objective functions is designed to obtain $\{\vec{r}_i, i=1, \ldots, n\}$ for specifying the constraint task for the redundant robot to fulfill. This approach allows the redundancy to be useful for other purposes, in addition to obstacle avoidance, by optimizing any other criterion. In application of the posture generation rule, each of link orientations would be determined by following the sequence starting from the end-effector toward the base. Consider the case of on-line posture planning. In order to consider the effect of the implicit path $\dot{\mathbf{x}}_i$ (by (13)) imposed on the end of each link $\mathbf{x}_i$, let

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \dot{\mathbf{x}}_i \times \dot{\mathbf{x}}_i \times \Delta t \tag{17}$$

where Δt is the sampling time.

4.1. Collision Avoidance

Each link orientation (or $\vec{r}_i$) is optimized by immobilizing the end $\tilde{\mathbf{x}}_{i+1}$ of link. This is equivalent to optimize the link vector $-\vec{O}_i$ with respect to the joint position (θ_i or (θ_i^x, θ_i^z) in 2-D or 3-D space) to make the link vector $-\vec{O}_i$ keep away from the obstacles as far as possible. Let $\{B_k\}$ be the set of obstacles cluttered in the workspace, and $\mathbf{b}_{ki}$ and $\mathbf{I}_{ik}$ denote the points on B_k and the i^{th} link, respectively, with the minimum distance $\|\mathbf{b}_{ki} - \mathbf{I}_{ik}\|$ between them. As Figure 3a, the vector $-\vec{O}_i$ would be optimized away from the collision vector $\vec{B}_{ki}$ on the polygonal obstacle $\mathbf{A}$, where $\vec{B}_{ki} = \mathbf{b}_{ki} - \tilde{\mathbf{x}}_{i+1}$. In order to maximize the included angles between $-\vec{O}_i$ and all of $\vec{B}_{ki}$, the following objective function Γ_1 is designed for the motion of joint θ_i (or joints θ_i^x and θ_i^z in 3D case).

$$\Gamma_1(\theta_i) = \sum_k \Gamma_1^{ik}(\theta_i, \vec{B}_{ki}),$$

where

$$\Gamma_1^{ik} = \begin{cases} 1, & d(B_k, \tilde{\mathbf{x}}_{i+1}) > \|\vec{O}_i\| \text{ or } \dfrac{-\vec{O}_i \cdot \vec{B}_{ki}}{\|\vec{O}_i\| \times \|\vec{B}_{ki}\|} < \cos\phi_i. \\ \left\{1 - \left[\left(\dfrac{-\vec{O}_i \cdot \vec{B}_{ki}}{\|\vec{O}_i\| \times \|\vec{B}_{ki}\|} - \cos\phi_i\right) \Big/ (1 - \cos\phi_i)\right]^2\right\}^{1/2}, & \text{otherwise} \end{cases} \tag{18}$$

In (18), $d(B_k, \tilde{\mathbf{x}}_{i+1})$ denotes the minimum *Euclidean* distance between the obstacle B_k and the end of link $\tilde{\mathbf{x}}_{i+1}$. In Figure 3a, the dashed orientation cone indicates the allowable range characterized in (18) in which the link vector $-\vec{O}_i$ would be moved away if any obstacle intersects this cone. This cone is characterized by specifying the included angle ϕ_i and the link length $\|\vec{O}_i\|$. It is the region of allowable cw and ccw rotation angles of ith link as the (i+1)th joint is fixed.

Let $\Psi_{ik} = \cos^{-1}(-\vec{O}_i \cdot \vec{B}_{ki} / \|\vec{O}_i\| \times \|\vec{B}_{ki}\|)$. A profile plotted by Γ_1^{ik} versus Ψ_{ik} is shown in Figure 3b, where the maximum value of Γ_1^{ik} is one if Ψ_{ik} is larger than ϕ_i. Furthermore, the gradient of Γ_1^{ik} would be larger while Ψ_{ik} is smaller, so that i^{th} joint velocity driving the link vector $-\vec{O}_i$ away from the obstacle $\vec{B}_{ki}$ is larger. To obtain the gradient of Γ_1^{ik} if any obstacle intersects the orientation cone of the i^{th} link, we have

$$\frac{d\Gamma_1^{ik}}{d\theta_i} = \frac{-\left(\dfrac{-(d\vec{O}_i/d\theta_i)\cdot\vec{B}_{ki}}{\|\vec{O}_i\|\times\|\vec{B}_{ki}\|}\right)\cdot\left(\dfrac{-\vec{O}_i\cdot\vec{B}_{ki}}{\|\vec{O}_i\|\times\|\vec{B}_{ki}\|} - \cos\phi_i\right)}{(1-\cos\phi_i)^2} \times \left\{1 - \left[\left(\frac{-\vec{O}_i\cdot\vec{B}_{ki}}{\|\vec{O}_i\|\times\|\vec{B}_{ki}\|} - \cos\phi_i\right) \Big/ (1-\cos\phi_i)\right]^2\right\}^{1/2}. \tag{19}$$

Secondly, the optimization of each link orientation would, if possible, also try to keep the Euclidean distance between the adjacent $(i-1)^{\text{th}}$ link and the set of obstacles $\{B_k\}$ as large as possible. To this aim, let $\vec{\rho} = \mathbf{b}_{k(i-1)} - \mathbf{I}_{(i-1)k}$ denote the directed distance vector from the i^{th} link toward the obstacle B_k. Define

$$D_{ik} = -\vec{O}_i + \vec{\rho} + \frac{\vec{\rho}\cdot\dot{\mathbf{x}}_{i+1}\cdot\vec{\rho}}{\|\vec{\rho}\|^2}, \tag{20}$$

where the last term of (20) $\vec{\rho}\cdot\dot{\mathbf{x}}_{i+1}\cdot\vec{\rho}/\|\vec{\rho}\|^2$ denotes the projection of $\dot{\mathbf{x}}_{i+1}$ onto the vector $\vec{\rho}$. Figure 4a shows an illustration for $\vec{D}_{ik}$, where $\vec{D}_{ik}$ is a reference point for representing the obstacle B_k. Therefore, $\vec{D}_{ik}$ is temporarily treated as a constant in the following objective function.

$$\Gamma_2(\theta_i) = \sum_k \Gamma_2^{ik}(\theta_i, \vec{D}_{ik}),$$

where

$$\Gamma_2^{ik} = \begin{cases} 1, & \|-\vec{O}_i - \vec{D}_{ik}\| > \eta. \\ \dfrac{2\eta}{\pi} \times \sin\left(\dfrac{\|-\vec{O}_i - \vec{D}_{ik}\|}{\eta} \times \dfrac{\pi}{2}\right), & \text{otherwise.} \end{cases} \tag{21}$$

Let $\tilde{D}_{ik} = -\vec{O}_i - \vec{D}_{ik}$ for convenience. In (21), $\|\tilde{D}_{ik}\|$ is the estimated distance between the obstacle B_k and the $(i-1)^{\text{th}}$

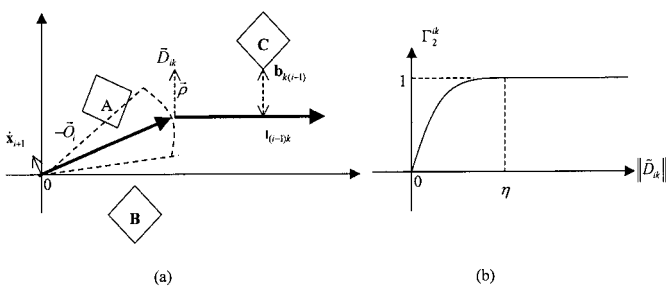

Fig. 4. (a) Moving a link should consider the distance away from the surrounding obstacles. (b) the plot of the objective function Γ_2^{ik} versus the estimated distance $\|\tilde{D}_{ik}\|$ between the adjacent $(i-1)^{\text{th}}$ link and an obstacle **C**.

link, and η indicates the desired safe distance between the link and each obstacle. As Figure 4b shows, the function Γ_2^{ik} reaches a maximum value 1 when the link i is far from the obstacle k. Furthermore, the gradient of Γ_2^{ik} is larger while $\|\tilde{D}_{ik}\|$ is smaller, which could be computed by

$$\frac{d\Gamma_2(\theta_i)}{d\theta_i}=\sum_k \frac{d\Gamma_2^{ik}(\theta_i,\vec{D}_{ik})}{d\theta_i},$$

where

$$\frac{d\Gamma_2^{ik}}{d\theta_i}=\frac{d(-\vec{O}_i)}{d\theta_i}\cdot\frac{\tilde{D}_{ik}}{\|\tilde{D}_{ik}\|}\times\cos\left(\frac{\|\tilde{D}_{ik}\|}{\eta}\times\frac{\pi}{2}\right). \quad (22)$$

4.2. Compliance of Joint Motion with Joint Limits

In addition to the avoidance of obstacles, the moving range of each revolute joint is also narrowed by its joint limit in angle to avoid disruption. The angle of each joint is equivalent to the difference of angular positions of two adjacent link vectors connected by the joint. Therefore, the joint positions of both θ_i and θ_{i+1} are moved correspondingly due to a change of the link orientation $\vec{O}_i$. Let $\{\theta_i^J, J=1, 2\}$ and $\{\theta_{i+1}^J, J=3, 4\}$ denote the (upper and lower) limits of joints θ_i and θ_{i+1}, respectively. For representing the internal restrictions, the orientation of ith link is bounded by four vectors as a function of joint angle and its upper or lower limit at both ends

$$\vec{B}_i^J=\begin{cases}-\left[\prod_{j=1}^{i-1}\mathbf{R}(\theta_j)\right]\mathbf{R}(\theta_i^J)\vec{L}_i, & \text{for } J=1, 2\\ -\left[\prod_{j=1}^{i+1}\mathbf{R}(\theta_j)\right]\mathbf{R}(\theta_{i+1}^J)^{-1}\vec{L}_i, & \text{for } J=3, 4\end{cases}. \quad (23)$$

Conceptually similar, avoiding a joint limit can be thought as to avoid a (virtual) obstacle having the collision vector $\vec{B}_i^J$. Therefore, the objective function $\Gamma_3(\theta_i)$ for avoiding joint limits of θ_i can be handled in the same way as (18), with defining $\vec{B}_{ki}$ and ϕ_i in (18) replaced by $\vec{B}_i^J$ and θ_i^J, respectively. Note that the objective function value of $\Gamma_3(\theta_i)$ is higher as the joints near their internal boundaries.

Remark. For the 3D manipulator specified in subsection 3.1, (23) can be revised as follows.

$$\vec{B}_i^J=\begin{cases}-\left[\prod_{j=1}^{i-1}\mathbf{R}(\theta_j^x, \theta_j^z)\right]\mathbf{R}(\theta_i^{x^J}, \theta_i^z)\vec{L}_i, & \text{for } J=1, 2\\ -\left[\prod_{j=1}^{i+1}\mathbf{R}(\theta_j^x, \theta_j^z)\right]\mathbf{R}(\theta_{i+1}^{x^J}, \theta_{i+1}^z)^{-1}\vec{L}_i, & \text{for } J=3, 4\end{cases}. \quad (24)$$

where $\theta_j^{x^J}$ denotes the limit of joint θ_j^x, namely, only θ_j^x has to concern the joints limits for the specified manipulator.

4.3. Connection of Posture

The optimization of link orientation should also minimize the displacements of joints since too large displacement may cause the links 1 and 2 disconnect with other links. A physically feasible planned posture should satisfy the connection constraint that the links should connect the end-effector to the base. However, sequentially orienting the links from the desired end-effector position to the base can't guarantee the connection of links, because each of the link orientations is independently determined. Therefore, minimization of (13) is performed in optimizing link orientations when any of links is free from collisions or joint limits. By (13), minimizing $\dot{\mathbf{x}}_i$ amounts to minimize

$$\|\dot{\mathbf{x}}_{i+1}+\mathbf{J}_i\dot{\theta}_i\|, \quad (25)$$

where $\mathbf{J}_i=d(-\vec{O}_i)/d\theta_i\in R^{m\times 1}$. Solving $-\dot{\mathbf{x}}_{i+1}=\mathbf{J}_i\cdot\dot{\theta}_i$ by least squares method, we obtain the following joint velocity for minimizing (25)

$$\theta_i^C=(\mathbf{J}_i^T\mathbf{J}_i)^{-1}\mathbf{J}_i^T\cdot-\dot{\mathbf{x}}_{i+1}. \quad (26)$$

4.4. Imposing Preference of Posture

In performing a specific task, the robotic posture is desired to be, say, as straight as possible to reach the constrained workspace deeply and then pull out of the end-effector from inside more easily. Such a tendency of posture generation is called *elastic*. To achieve certain degree of elasticity in orienting a link to avoid self-collisions among links, the desired orientation for each link is defined by

$$\vec{E}_i=\frac{\vec{O}_{i+1}}{\|\vec{O}_{i+1}\|}+\frac{\vec{O}_{i-1}}{\|\vec{O}_{i-1}\|}. \quad (27)$$

Namely, the desired link vector $\vec{E}_i$ is the bisector of two adjacent link vectors. The objective function for this goal is

$$\Gamma_4(\theta_i,\vec{E}_i)=\frac{\vec{O}_i\cdot\vec{E}_i}{\|\vec{O}_i\|\times\|\vec{E}_i\|}. \quad (28)$$

Because the vector $\vec{E}_i$ depends also on neighboring link orientations, the movement imposed on any link to avoid obstacles would be propagated among the adjacent links of this link. Therefore, a manipulator configuration that optimizes (28) tends to a more straight posture while reaching inside of and then out of the constrained workspace modeled by a union of convex polyhedra or line segments.

4.5. Priorities of the Optimizations

With multiple objectives, a compromise is made to achieve a better performance by setting priorities among the goals. The importance of avoiding obstacles and joint limits are set prior to achieving the connective and preferred posture for

posture determination. A combined objective function for each $\vec{r}_i$ is designed as

$$\vec{r}_i = \sum_j \Phi^j \times \frac{d(-\vec{O}_i/\|\vec{O}_i\|)}{d\theta_i^j},$$

where the summation index $j=z$ for a planar robot and $j=x, y$ for a spatial robot

$$\Phi^j = \omega\left(\gamma_i^1 \frac{d(\Gamma_1+\Gamma_2+\Gamma_3)}{d\theta_i^j} + (1-\gamma_i^1) \times \left[\frac{\gamma_i^2}{\omega}\theta_i^C + (1-\gamma_i^2)\frac{d\Gamma_4}{d\theta_i^j}\right]\right), \quad (29)$$

where $0 \le \gamma_i^1, \gamma_i^2 \le 1$ determine the priorities among the objective functions, and ω is the parameter specified by user. Note that γ_i^1 provides higher priority for the avoidance of obstacles and joint limits than for the others and γ_i^2 makes the connective posture prior to the elastic posture, which could be defined by

$$\gamma_i^1 = \max_{k,J}\left\{\left\{\frac{-\vec{O}_i \cdot \vec{B}_{ki}}{\|\vec{O}_i\| \times \|\vec{B}_{ki}\|}\right\}, \left\{\cos\left(\frac{\|\tilde{D}_{ik}\|}{\eta} \times \frac{\pi}{2}\right)\right\}, \left\{\frac{-\vec{O}_i \cdot \vec{B}_i^J}{\|\vec{O}_i\| \times \|\vec{B}_i^J\|}\right\}\right\}, \quad \gamma_i^2 = 1 - \exp\left(-\frac{\|\dot{\mathbf{x}}_{i+1}\|}{\delta}\right) \quad (30)$$

where δ specifies the gradient of the function γ_i^2 that depends on the disconnection $\dot{\mathbf{x}}_{i+1}$ between the i^{th} and $(i+1)^{\text{th}}$ links.

5. SIMULATION EXPERIMENTS

The motion of the robots in the following simulations will illustrate the performance of multi-criteria based approach. For simulations, the links of redundant robots are modeled by a union of line segments or convex polyhedra. The Euler integration of the joint velocity (9) would be used to update the joint positions. The constraint task $\mathbf{J}_C^* \dot{\mathbf{x}}_C$ would be specified by using the Equations (14), (15), (16), and (29) The obstacles cluttered in the workspace are composed of the convex polygons or polyhedra. Besides, the GJK algorithm[31] for computing the closest points and minimum distance between two convex polyhedra would be employed to obtain the distance parameters used in (17) to (20) (i.e. $d(B_k, \tilde{\mathbf{x}}_{i+1})$, $\mathbf{b}_{ki}$, and $\mathbf{I}_{ik}$). In the following two simulations, the end-effector trajectory planning strategy is (reference 2): Plan a series of "Key Path Points" along the path of end-effector off-line. "Key Path Points" means path points at which the links collide with the obstacles presumably or where the direction of motion substantially changes. The points are connected by straight lines. For successive points along the specified end-effector path, the collision-free postures are resolved from collision-free link orientations derived by the proposed technique.

Simulation 1. Figure 5a shows a constrained workspace enclosed by seven rectangles for obstacle avoidance experiment and a planar 12 dof redundant manipulator fixed at the base (9, 4). The lengths of links are specified as [2.2, 2.0, 2.0, 1.8, 1.8, 1.6, 1.6, 1.4, 1.4, 1.2, 1.2, 1.0] and the initial joint position is [–10°, –25°, 40°, 50°, 65°, 80°, –109°, 50°, 70°, –20°, –30°, 20°]^T. The path of the end effector is shown by the dashed line inside the workspace for performing the task of end effector into and then out of the constrained workspace. The velocity $\dot{\mathbf{x}}_E$ of the end effector is uniform with the motion 0.1 for each cycle. To specify the parameters, $\eta=1$, $\phi_i=\tan^{-1}(\eta/\|\vec{O}_i\|)$, $\omega=0.1$, $\delta=0.3$, and

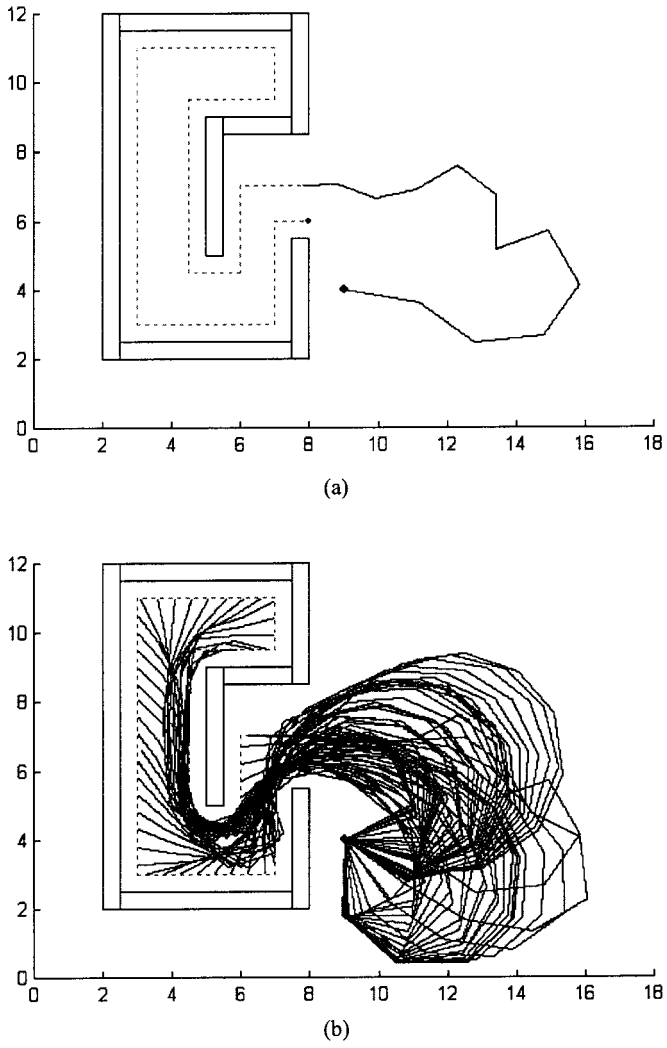

Fig. 5. Simulation result: (a) shows the path (dashed line) in a constrained workspace consisting of seven rectangle polygons and a planar manipulator fixed at the base (9, 4) with 12 degrees of freedom; (b) shows the strobed version of postures when the end-effector tracked the path. Namely, the planar manipulator could successfully put the links into and then out of the workspace.

$\{\theta_i^J = 135°$ and $\phi_i^J = 55°, \forall J$ and $\forall i\}$. Based on the proposed rule, the links orientations of a posture would be optimized one by one from the end effector to the base. One should be noted that the objective function Γ_4 for the elastic posture would not be applied to the optimization of the first link in the simulation in order to obtain the more connective posture. Figure 5b evidences good performance of the simulation result, in which strobed trajectory of the generated postures is plotted. Due to imposing the preference of postures, the final collision-free postures for each movement trends to be straight. It is seen that the planar manipulator could successfully navigate the links into and then out of the workspace without collisions.

Simulation 2. For simulation in the 3-D workspace, the redundant manipulator is required to maneuver the links through the window, which obstructs the motion of outer links (forearms), without collision of its links with the edges of the window. It is shown in Figure 6a, where the window is modeled by polyhedra. It is assumed that the window vertices lie on a plane and are stationary at a specified position and orientation. The manipulator is constructed by 9 links (including the link of base) and 16 degrees of freedom, whose prototype has been described in the subsection 3.1. Note that each of the links in fact consists of two parts connected by the joint θ_i^z. The lengths of links are specified as [3.0, 2.4, 2.2, 2.0, 1.8, 1.6, 1.4, 1.2, 1.0] and the initial joint position is $[-\pi, 1.5, 0, -0.8, 0, -0.8]$. The obstacles cluttered in the workspace are composed of the convex polygons or polyhedra. 0, –0.6, 0, –0.5, 0, –0.5, 0, –0.3, 0, –0.3]^T (rad). The tracking path of the end effector consists of five connecting line segments: start from the center of window to track the contour of the obstacle behind the window and then return to this center. The velocity $\dot{\mathbf{x}}_E$ between the ends of each line segment of the path is interpolated using a fifth-order polynomial time law such that null initial and final velocities and accelerations are obtained. To specify the parameters, $\eta = 1.2$, $\phi_i = \tan^{-1}(\eta / \|\vec{O}_i\|)$, $\omega = 0.2$, $\delta = 0.3$, and $\{\theta_i^{xJ} = 135°$ and $\phi_i^J = 55°, \forall J$ and $\forall i\}$. One should be noted that each link vector $\vec{O}_i$ denotes the central axis of link and η should be specified larger in order to reflect the volume of link in 3D case. In addition, the objective function Γ_4 is not applied to the optimization of the first link in the simulation for obtaining more connective posture. Figure 6b to Figure 6d show the same posture trajectory of the simulation result via different viewpoints.

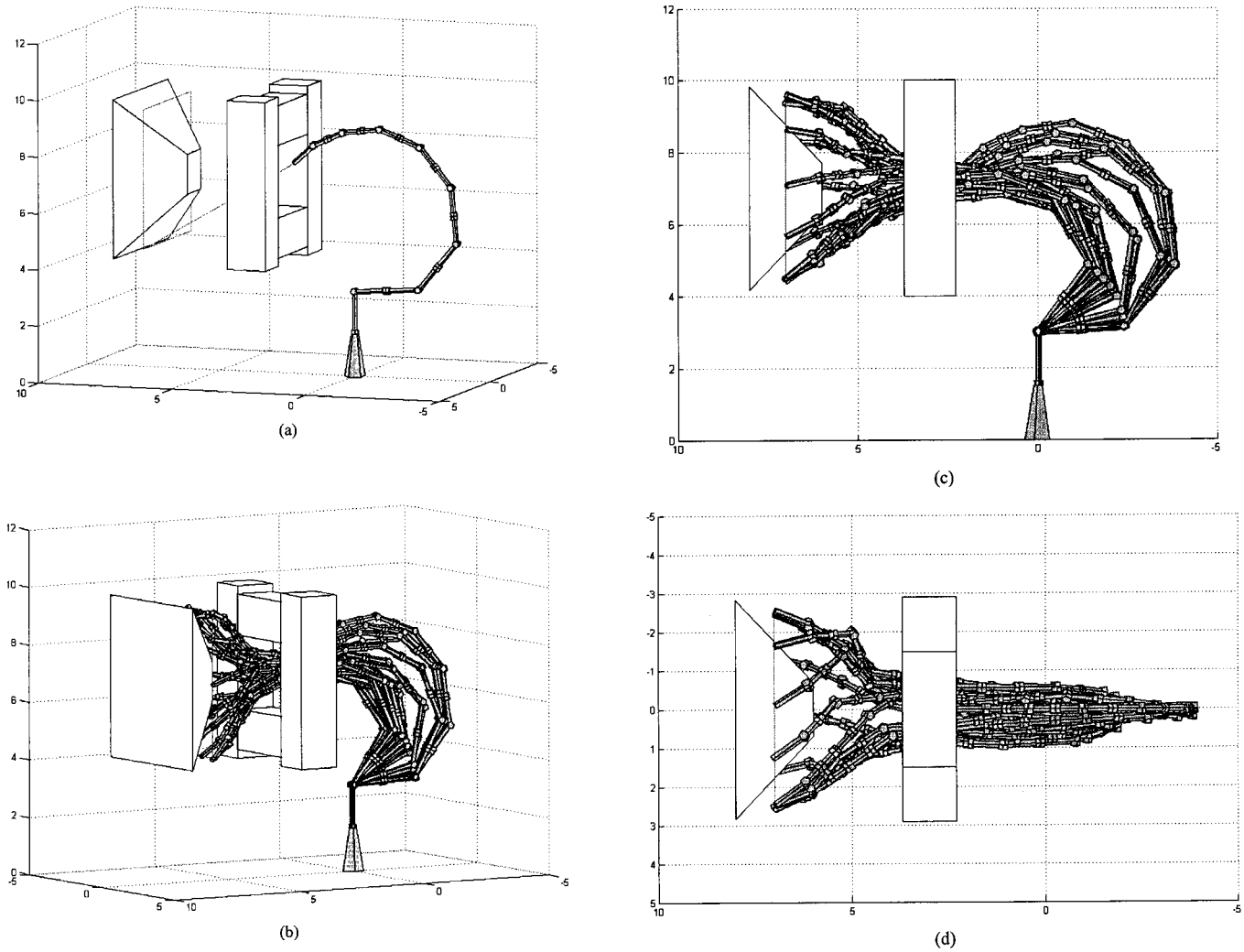

Fig. 6. Simulation result: (a) a redundant manipulator with 9 links (including the link of base) and 16 degrees of freedom has to maneuver the links through the window. Then, (b), (c), and (d) show posture trajectory from different views.

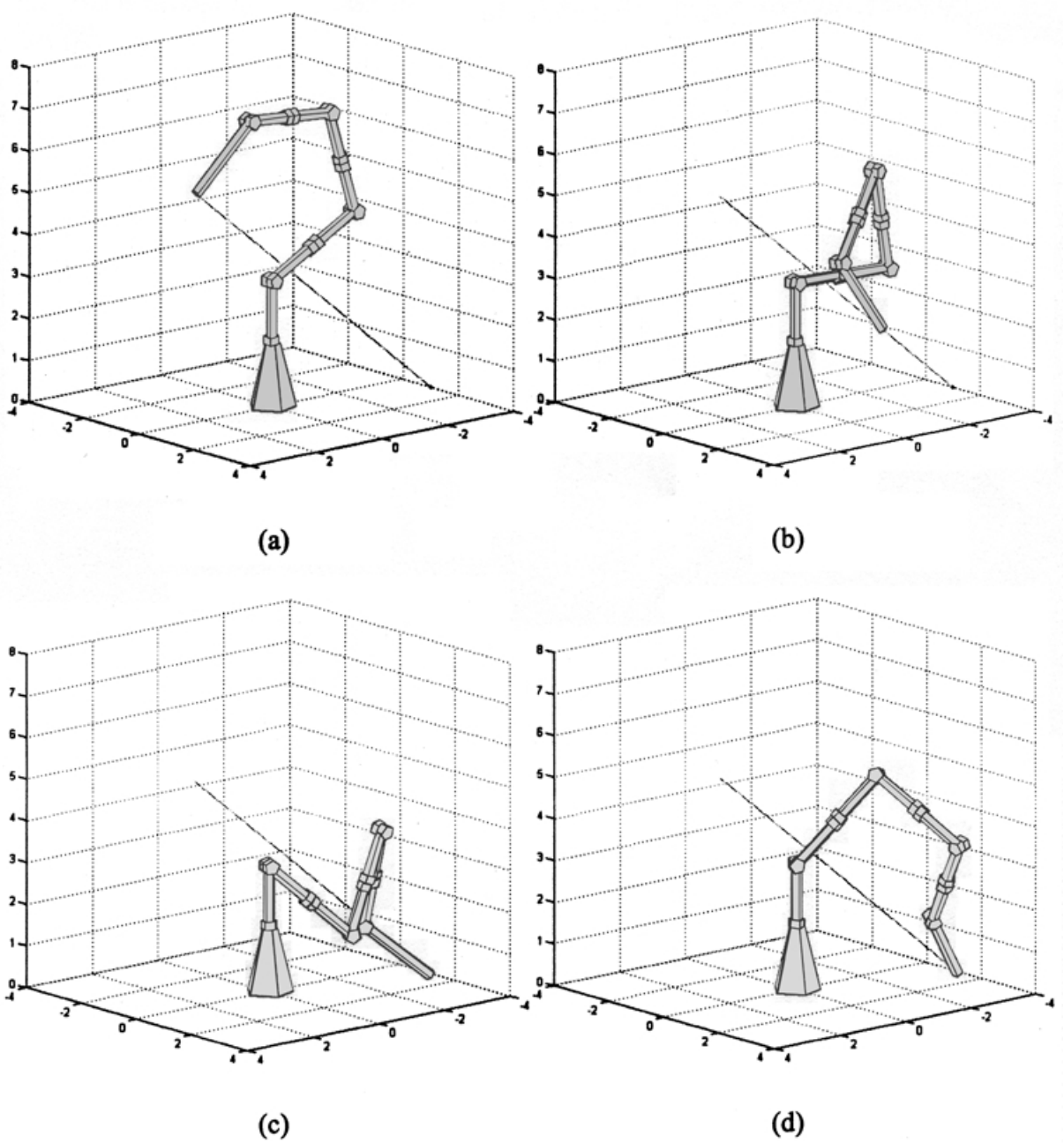

Fig. 7. Simulation results: (a) shows a redundant manipulator with 5 links and 8 degrees of freedom. In (b), the limit of the joint 6 between the third and fourth links has been reached, then (c) the impossible posture of the manipulator is obtained finally. On the other hand, (d) shows the final result concerning the joint limits in which the links obviously made a turn in the posture.

From the snapshots of simulations, the links avoid collisions with the window edges while the manipulator maneuvers its end-effector into and out of the window frame.

Simulation 3. A simulation for demonstrating the avoidance effect of joint limits is performed in the following. Figure 7a shows a redundant manipulator with 5 links and 8 degrees of freedom. The lengths of links are specified as [3.0, 2.4, 2.2, 2.0, 1.8]. The path of the end-effector is to move on the shown line segment started from the initial configuration at $[-\pi, 1.1, 0, -1.4, 0, -1.2, 0, -0.8]^T$ (rad) toward the goal position at (–4,1, 0) on the ground (with an arbitrary configuration). The links have to make a great turn in the posture for tracking the specified straight-line path. The end-effector velocity $\dot{\mathbf{x}}_E$ on the path is interpolated using a fifth-order polynomial time law such that null initial and final velocities and accelerations are obtained. For comparison, two cases are performed: one uses the minimum-norm inverse kinematics control, and the other case uses the multiple objective functions as the optimization critera with the parameters, ω=0.2, γ_i^J=0, δ=0.3, and $\{\theta_i^{x^J}$=135° and ϕ_i^J=55°, $\forall J$ and $\forall i\}$. For the first case, Figure 7b shows that the limit of the joint 6 connecting the third and fourth links has been reached when using minimum-norm inverse kinematics for control. Figure 7c shows an impossible configuration of the manipulator, which requires the dis-assemble and re-assemble of joints. Compared to Figure 7c, Figure 7d shows the final result

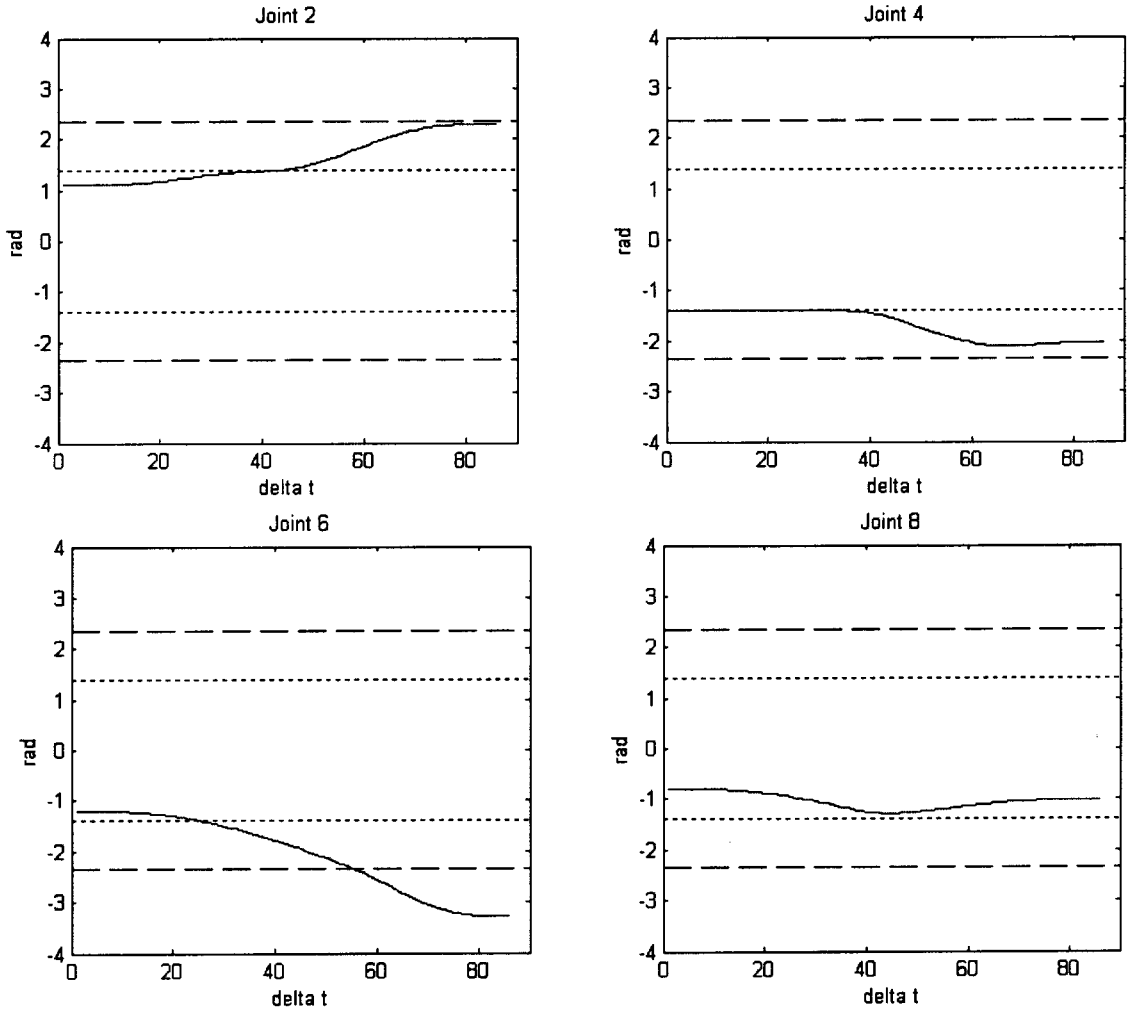

Fig. 8. Referring to the simulation in Figure 7c, some of the joints exceed or are very close to the limit at 2.36 (rad) that is denoted by the long dashed line because the joint limits are not imposed.

concerning the joint limits in which the links obviously make a great turn in the posture. Moreover, Figure 8 and Figure 9 show the evolution of the joint positions for the two comparative simulations. Some joints shown in Figure 8 are very close to or violate the limit at 2.36 (rad) (denoted by the long dashed line in Figure 8) because the limits of joints are not activated. On the contrary, the joints shown in Figure 9 could be naturally driven away from their respective joint limits while the joint positions entered the influence range 1.40 to 2.36 (rad).

6. CONCLUSION

In this paper, we have presented a posture generation rule and an efficient gradient projection scheme for an objective function integrating multiple criteria to generate the singularity-robust null-space projection vector. The rule reveals that the end of each link has to track an implicit path that is indirectly resulted from the constrained (trajectory tracking) motion specified for the end-effector. Therefore, each of link orientations is sequentially determined from the end-effector toward the base by optimizing multi-criteria. The technique could optimize each link orientation by following the gradient of a combination of multiple objective functions incorporating the collision avoidance, the joint limit, the posture connection, and the elastic posture simultaneously to realize a compromise between the primary and secondary tasks. Compared to the other techniques, the proposed multi-criteria based scheme for singularity-robust null-space projection vector is developed for fulfilling multiple secondary objectives without interfering the primary task as much as possible. The major advantage of the technique is that the computation load of on-line posture planning can be reduced, since the posture is link-by-link sequentially computed, and each link orientation requires only one computation within each computing cycle. However, the secondary task tracking may be less accurate. Accordingly, this rule can be effectively applied to redundant manipulators with a larger degree of freedom, where the solution space is large. Simulations have been performed for (i) a planar redundant manipulator into and then out of a constrained workspace and (ii) a 3D redundant robot through a window without collision with the edges to validate the effectiveness and flexibility of the multi-criteria based method for generating proper postures as null-space projection vector.

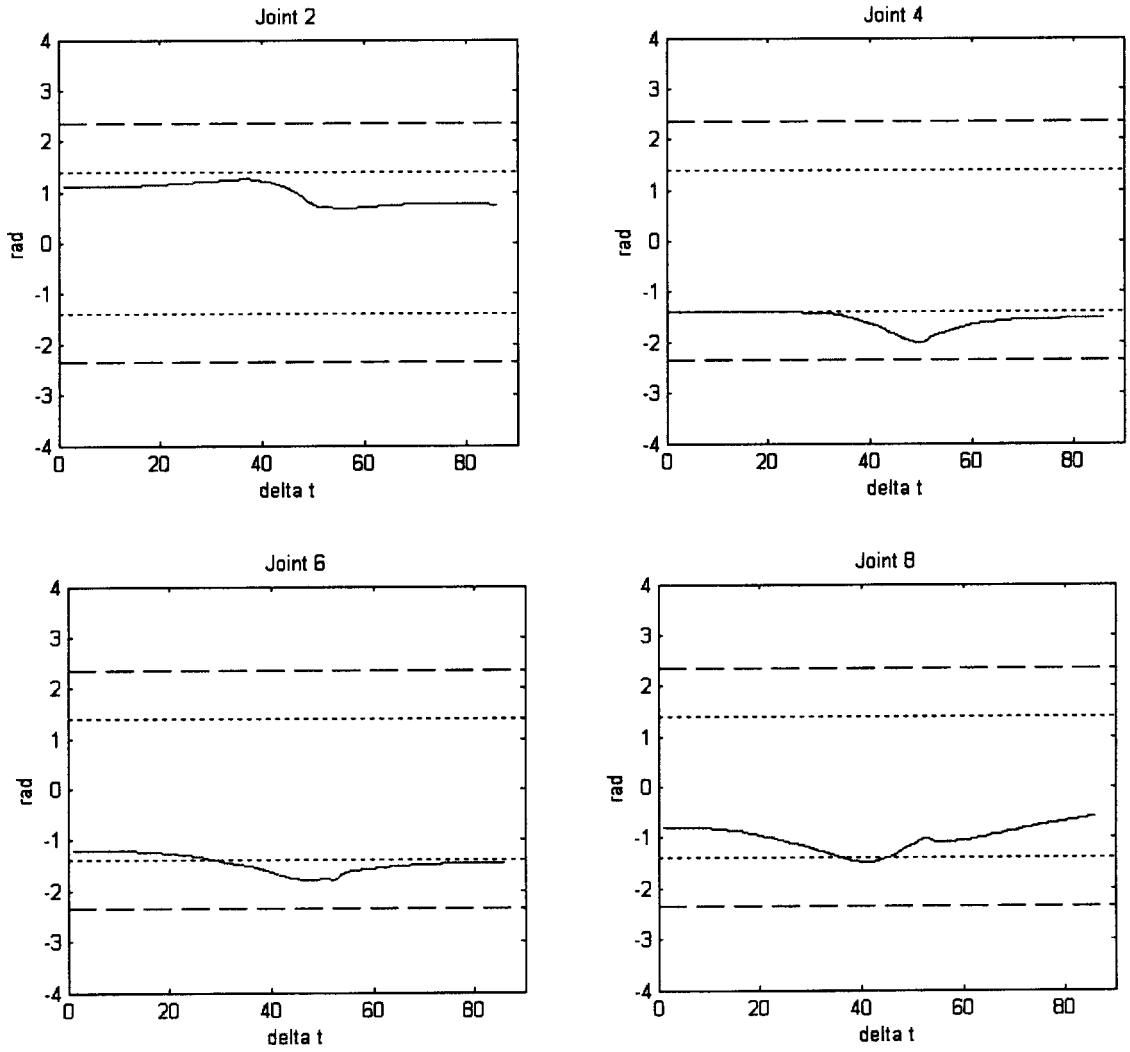

Fig. 9. As the joint limits are imposed in Figure 7d, the joints could be driven away from the joint limit at 2.36 (rad) while their positions entered the influence range 1.40 to 2.36 (rad).

ACKNOWLEDGEMENT

This research was supported by National Science Council of R.O.C. under contract NSC 89–2212-E–001–002.

References

1. J. A. Kuo and D. J. Sanger, "Task planning for serial redundant manipulators", *Robotica* **15**, Part 1, 75–83 (1997).
2. T. C. Liang and J. S. Liu, "An improved trajectory planner for redundant manipulators in constrained workspace", *J. of Robotic Systems* **16**(6), 339–351 (1999).
3. E. S. Conkur and R. Buckingham, "Manoeuvring highly redundant manipulators", *Robotica* **15**, Part 1, 435–447 (1997).
4. J. Z. Li and M. B. Trabia, "Adaptive path planning and obstacle avoidance for a robot with a large degree of redundancy", *J. of Robotic Systems* **13**(3), 163–176 (1996).
5. M. T. H. Beheshti and A. K. Tehranl, "Obstacle avoidance for kinematically redundant robots using an adaptive fuzzy logic algorithm", *Proc. the American Control Conference* (1999) pp. 1371–1375.
6. A. McLean, and S. Cameron, "The virtual springs method: path planning and collision avoidance for redundant manipulator", *Int. J. of Robotics Research* **15**(4), 300–319 (1996).
7. K. Glass, R. Colbaugh, D. Lim and H. Seraji, "Real-time collision avoidance for redundant manipulators", *IEEE Trans. On Robotics and Automation* **11**(3), 448–457 (1995).
8. W. S. Newman, "Automatic obstacle avoidance at high speeds via reflex control", *Proc. IEEE Int. Conf. Robotics and Automation* (1989), pp. 1104–1109.
9. T. S. Wikman and W. S. Newman, "A fast, on-line collision avoidance method for a kinematically redundant manipulator based on reflex control", *Proc. IEEE 7nt. Conf. Robotics and Automation* (1992) pp. 261–266.
10. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *Int. J. of Robotics Research* **5**(1), 90–98 (1986).
11. Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based on redundancy control of robot manipulators", *Int. J. of Robotics Research* **6**(2), 3–15 (1987).
12. S. I. Choi and B. K. Kim, "Obstacle avoidance control for redundant manipulators using collidability measure", *Robotica* **18**, 143–151 (2000).
13. S. I. Choi and B. K. Kim, "Obstacle avoidance for redundant manipulators using directional-collidability/temporal-collidability measure", *Journal of Intelligent and Robotic Systems* **28**(3), 213–229 (2000).
14. Z. Y. Guo and T. C. Hsia, "Joint trajectory generation for redundant robots in an environment with obstacles", *Proc. IEEE Int. Conf. Robotics and Automation* (1990), pp. 157–162.
15. N. Rahmanian-Shahri and I. Troch, "Collision-avoidance control for redundant articulated robots", *Robotica* **13**, 159–168 (1995).

16. N. Rahmanian-Shahri and I. Troch, "A new on-line method to avoid collisions with links of redundant articulated robots", *Robotica* **14**, 611–619 (1996).
17. A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", *Int. J. of Robotics Research* **4**(3), 109–117 (1985).
18. W. J. Cho and D. S. Kwon, "A sensor-based obstacle avoidance for a redundant manipulator using a velocity potential function", *Proc. IEEE Int. Workshop on Robot and Human Communication* (1996), pp. 306–310.
19. J. Wunderlich and C. Boncelet, "Local optimization of redundant manipulator kinematics within constrained workspaces", *Proc. IEEE Int. Conf. Robotics and Automation* (1996) pp. 127–132.
20. C. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators", *IEEE Trans. On Systems, Man, and Cybernetics* **13**(3), 245–250 (1983).
21. S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulator", *IEEE Trans. On Robotics and Automation* **13**(3), 398–410 (1997).
22. D. Li, A. A. Goldenberg, and J. W. Zu, "Peak torque reduction with redundant manipulators", *Proc. IEEE Int. Conf. Robotics and Automation* (1996), pp. 1775–1780.
23. J. M. Hollerbach and K. I. C. Suh, "Redundancy resolution of manipulators through torque optimization", *IEEE J. of Robotics and Automation* **3**, 308–316 (1984).
24. R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators", *IEEE Trans. On Robotics and Automation* **12**(4), 543–552, 1996.
25. O. Egeland, 'Task space tracking with redundant manipulators", *IEEE Trans. Robotics and Automation* **3**(5), 471–475 (1987).
26. J. Barraquand and J. C. Latombe, "Robot motion planning: A distributed representation approach", *Int. J. of Robotics Research* **10**(6), 628–649 (1991).
27. J. C. Latombe, "*Robot motion planning*", Kluwer Academic Publishers, 1991.
28. T. Lozano-Pérez, "Spatial planning: A configuration space approach", *IEEE Trans. On Computers* **32**(2), 108–120 (1983).
29. H. Seraji and R. Colbaugh, "Improved configuration control for redundant robots", *J. of Robotics Systems* **7**(6), 897–928 (1990).
30. C. W. Wampler II, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares method", *IEEE Trans. On Systems, Man, and Cybernetics* **16**, 93–101 (1986).
31. E. G. Gilbert, D. W. Johnson and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space", *IEEE Trans. Robotics and Automation* **4**, 193–203 (1988).