On-Line Posture Generation of Redundant Manipulator for Collision Avoidance In Dynamic Environments

Jin–Liang Chen and Jing–Sin Liu
Institute of Information Science 20, Academia Sinica, Nankang, Taipei 115, Taiwan, R.O.C.,
Email: casper@iis.sinica.edu.tw and liu@iis.sinica.edu.tw

Abstract. The redundant manipulator has the advantage to generate the same configuration of the end-effector with infinite number of joint motion, which increases dexterity and versatility for performing multi-task at the end-effector and self-motion of robotic body. In this report, we propose a principle useful for on-line globally planning postures of the redundant manipulator. For obstacle avoidance, a technique is proposed to on-line generate collision-free joint velocities for self-motion of robotic body based on the incorporation of the principle and the pseudoinverse control. Simulations were implemented in the 2D static and dynamic environments in which obstacles were represented as convex polygons, which verify the good performances of collision avoidance using the proposed technique.

Key word: collision avoidance, redundant manipulator, posture generation, on-line, and dynamic environment.

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 on the kinematic structure of a manipulator, which usually coincides with the number of independently controlled drives (i.e. joints). For example, a planar manipulator possessed three revolute joints has one redundant DOF for performing self-motion of the robot's body in the two-dimensional (2D) space while position of the end-effector (i.e. tip of the manipulator) is not affected. However, the robot is nonredundant for tasks involving both position and orientation of the end-effector. In the three-dimensional (3D) space, a nonredundant manipulator has six DOFs to position and orient the end-effector in any desired configuration. Therefore, a 3D manipulator is redundant if it possessed seven or more DOFs. Significantly, the redundancy in a manipulator structure is useful for avoiding obstacles [4-23], joint limits, and singularity of kinematic matrices [26-28], or peak toque reduction [29], toque optimization [30-32], and joint failure/fault tolerance [33-34], while the works performed at the end-effector. In other words, the extra DOF provides infinite number of joint motion to generate the same configuration of the end-effector, which increases dexterity and versatility for the robot.

Collision avoidance is a fundamental problem for the redundant manipulator, which guides the robot to avoid obstacles cluttered in the environment while performing the primary tasks at the end-effector. In practice, the problem can be solved either off-line or on-line planning. First, the high-level path planning traditionally indicates to globally find a collision-free path in the configuration space (i.e. joint space or the manipulator) before performing tasks [1-3]. 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 the low- dimensional workspace (i.e. 2 or 3) by previously planning collision-free configurations of the robot at many successive points of the end-effector following a given path [4,5], or by heuristically and frequently adjust the links of the robot from collision to obstacles [6-8] for planning collision-free trajectory. Furthermore, McLean and Cameron [9] proposed a virtual spring method for quickly 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 [9-23], which is focused on the problem of controlling a redundant robot so that the end-effector tracks a given path in the workspace as closely as possible and simultaneously ensures that the links avoid obstacles. Reasoned as above, such techniques naturally depend on the use of different control frameworks. For example, Glass et al. [10,11] proposed an approach that represents the collision avoidance requirements as a set of kinematic inequality constraints for implementations of the configuration control [24,25], and then ensures that these constraints are satisfied while tracking the given path. In [12], Newman proposed a concept of reflex control in order to interfere the control command given from a high-level planner to the controller 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-dimension) for guiding the interference right enough [13]. Based on the robot dynamic control in the operational space, Khatib [14] proposed a method using the artificial potential field in the operational space to generate joint toques directly from the force of potential gradient. Besides, the pseudoinverse of Jacobian matrix is a well-known technique for controlling kinematically redundant manipulators [26]. Generally, works performed at the end-effector is the primary task and utilization of the redundancy refers to the second task. The major advantage of using the pseudoinverse technique is that the second task can be executed through the null-space projection of pseudoinverse without affecting the primary task [15]. Thus, the technique is usually employed to optimize some specified objective functions for obstacle avoidance [16-23].

On the other hand, many techniques tried to represent the workspace by a mathematically analytic model because use of the gradient technique is possible for optimizing any objective function. In order to represent objects in the workspace with analytic equations, simple shapes of sphere, ellipse, or other surfaces are usually employed to represent obstacles or links of the robot. In [14], Khatib proposed analytic equations representing envelopes which best approximate the shapes of obstacle's primitives such as parallelepiped, finite cylinder, and cone. Furthermore, Choi and Kim [16] represented obstacles and links of the robot as spheres and ellipsoids, respectively. Therefore, this method predicts collision between the sphere and the ellipsoid via measuring their directional and temporal meeting [17], and then assigns escaped velocities to joints based on gradient of analytic function of the collision measure. Similarly, the method in [18] has to assume that the distance between each obstacle and each link is a differentiable function, i.e., an analytic equation. Besides, Rahmanian-Shahri and Troch 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 [19-20]. Although representing objects in analytic equations is useful for optimization of criteria, the free space for the robot maneuvering is considerably reduced, especially for narrow paths between objects. Even if objects were divided to approach the shapes of analytic models, the computational complexity depended on the number of objects would be increased.

In addition to the optimization of objective functions, Maciejewski and Klein [21] suggested the method that dynamically identifies the point on manipulator that is closest to an obstacle and assigns to it a desired velocity away from the obstacle surface based on the task-priority technique [15]. Thus, the method is applied to maneuver in the constrained workspace [22] without analytic models of objects. However, the velocities assigned to these obstacles avoidance points have to be specified heuristically, which may result in oscillation of robot’s links in narrow paths when magnitudes of the velocities are specified too large. In [23], an improved artificial potential field is utilized to specify the velocities away from obstacles, which has no local minima while all obstacles are convex and gaps exist between these obstacles.

In this report, we propose a principle useful for on-line globally planning postures of the redundant manipulator when the end-effector has to track a given path in workspace. A proper posture of the robot is depended on the task of collision avoidance. The principle suggests that the end of each link has to track an implicit path in the workspace, thus the orientations of links can be determined by a particular sequence. In order to generate the collision-free postures of links at every moment, a method is suggested for determining collision-free orientations of links for obstacle avoidance. Thus, a set of collision-free joint velocities can be resulted from the set of target and original orientations. Furthermore, the technique of pseudoinverse is utilized to control motion of the end-effector. Based on use of null-space projection of the pseudoinverse, a proper posture of the manipulator is approached without affecting the motion of the end-effector.

This report is organized as follows. First, the pseudoinverse kinematical control is briefly reviewed in Section 2. The principle of posture generation is introduced in Section 3. For determining the orientations of links based on the principle, a method is suggested for the 2D manipulator in Section 4. Moreover, further numerical simulations are shown in Section 5. Finally, conclusions are given in Section 6.

2. Pseudoinverse 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 $\mathbf{x_e}(t)\in\boldsymbol{R}^{\boldsymbol{m}}$, which is a function of time t and may be specified in advance or on-line determined with a joystick. The forward kinematics that map the joint vector $\boldsymbol{\theta}(t)\in\boldsymbol{R}^{\boldsymbol{n}}$ to $\mathbf{x_e}(t)$ can be represented as the function

$$\mathbf{x_e} = f(\boldsymbol{\theta}), \tag{1}$$

and then the forward rate kinematics are

$$\dot{\mathbf{x}}_{\mathbf{e}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \tag{2}$$

where $\boldsymbol{J}(\boldsymbol{\theta})\in\boldsymbol{R}^{\,m\times n}$ is Jacobian matrix of the end-effector [26]. For obtaining the joint rate velocity $\dot{\boldsymbol{\theta}}$, the general solution of Eq.(2) is

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

where $\boldsymbol{J}^{+}=\boldsymbol{J}^{T}(\boldsymbol{J}\boldsymbol{J}^{T})^{-1}$ denotes the pseudoinverse of $\boldsymbol{J}$, $\boldsymbol{I}\in\boldsymbol{R}^{n\times n}$ is the identity matrix, and $\mathbf{z}$ is an arbitrary vector in the joint velocity space which determines the use of manipulator redundancy. It is well known that $\boldsymbol{J}^{+}\dot{\mathbf{x}}_{\mathbf{e}}$ is the minimum-norm solution of $\|\dot{\boldsymbol{\theta}}\|$ and $(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})$ is the null-space projection matrix [26]. The homogeneous term $(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\mathbf{z}$ is orthogonal to $\boldsymbol{J}^{+}\dot{\mathbf{x}}_{\mathbf{e}}$, i.e., $(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})^{T}\boldsymbol{J}^{+}=0$, which results in self-motion of the manipulator body without affecting the motion $\dot{\mathbf{x}}_{\mathbf{e}}$ at the end-effector, i.e., $\boldsymbol{J}(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\mathbf{z}=0$. Therefore, such self-motion can be used to perform the second tasks mentioned in Section 1, e.g., obstacle avoidance.

For obstacle avoidance with the use of manipulator redundancy, the vector $\mathbf{z}$ can be implemented as a specified joint velocity toward a collision-free posture. The problem concerned in this report just finds such collision-free velocity vector $\mathbf{z}$ at every moment among a set of convex polygons cluttered in the workspace while the end-effector is tracking the given motion $\dot{\mathbf{x}}_{\mathbf{e}}$. Compared to the other methods [16-20], a novel principle is introduced for posture generation of the redundant manipulator in Section 3 and a method that is proposed based on use of the principle is without the analytic models to represent objects or the use of artificial potential field.

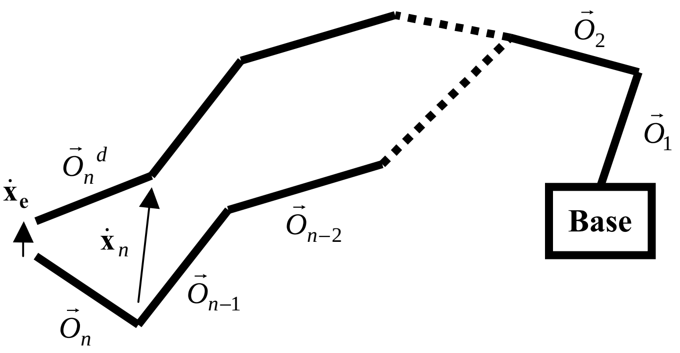

Fig.1. Posture of the last link is determined on $\dot{\mathbf{x}}_\mathbf{e}$ and the rotation of its orientation.

3. Posture Generation Principle

Assume the planar redundant manipulator with *n* degrees of freedom (i.e. *n* links), the end-effector has to track a specified path for performing technical tasks in workspace. Figure1 shows the end-effector at the n^{th} link should move with a desired velocity vector $\dot{\mathbf{x}}_\mathbf{e}$. Naturally, determining the posture of the n^{th} link only depends on its orientation. By definition, the orientation of the i^{th} link $\vec{O}_i = \mathrm{H}_{i+1} - \mathrm{H}_i$, where H_i denotes position of the i^{th} joint, for *i=1…n,* and H_{n+1} indicates position of the end-effector. Assume $\vec{O}_i{}^d$ is the desired orientation of the i^{th} link due to tasks, the motion of H_i resulted from rotating $\vec{O}_i$ to $\vec{O}_i{}^d$ when fixing at the H_{i+1} corresponds to

$$\vec{r}_i \;=\; \left(-\vec{O}_i{}^d\right)-\left(-\vec{O}_i\right). \tag{4}$$

Thus, the velocity $\dot{\mathbf{x}}_i$ resulted from the total motion of H_i is

$$\dot{\mathbf{x}}_i \;=\; \dot{\mathbf{x}}_\mathbf{e} \;+\sum_{j=i}^{n}\vec{r}_j\;, \tag{5}$$

which means that each of the links has to track an *implicit* path resulted from Eq.(5). In fact, Eq.(5) can be reformulated as the recursive form

$$\dot{\mathbf{x}}_i \;=\; \dot{\mathbf{x}}_{i+1} + \vec{r}_i\,, \tag{6}$$

Importantly, this result indicates the fact that posture of the redundant manipulator can be determined through specifying the set of desired orientations $\{\vec{O}_i^{\,d}$, i=1…$n\}$. Therefore, the problem for generating the posture can be transformed from searching in the joint space into determining the orientation vectors in the workspace. Actually, directly determining the collision-free orientation vectors to obstacles in workspace is easier than searching the collision-free joint vector in joint space.

To determine the set of orientation vectors $\{\vec{O}_i^{\,d}$, i=1…$n\}$, distribution of obstacles around these links should be concerned, which was discussed in Section 4. Furthermore, $\{\vec{O}_i^{\,d}$, i=1…$n\}$ should be determined backward from n th to (n – 2)th link because of the fact in Eq.(6). Noted that the *1*st and *2*nd links has only one solution for maintaining the immobile base.

Assume the set $\{\vec{O}_i^{\,d}$, i=1…$n\}$ has been determined for the obstacle avoidance. To calculate the collision-free joint vector $\boldsymbol{\theta}^d$= $[\theta_1^{\,d} \ldots \theta_i^{\,d} \ldots \theta_n^{\,d}]^T$ from $\{\vec{O}_i^{\,d}$, i=1…$n\}$, each of joints can be solved as follows. First, the relationship between the joint vector $\boldsymbol{\theta}$=$[\theta_1 \ldots \theta_i \ldots \theta_n]^T$ and the set of $\{\vec{O}_i$, i=1…$n\}$ is

$$\vec{O}_i = \tilde{R}_i \vec{L}_i = \left(\prod_{j=1}^{i} R(\theta_j)\right)\vec{L}_i\,,\ \text{ for } i=1\ldots n, \tag{7}$$

where $\tilde{R}_i$ denotes the synthetic rotation matrix from $\vec{L}_j$ to $\vec{O}_i$, $\{\vec{L}_i$, *i=1…n* $\}$ is the set of orientations when $\boldsymbol{\theta}$=0, and $R(\theta_j)$ is the rotation matrix with respect to θ_j. In order to solve the joint angle θ_i between $\vec{O}_{i-1}$ and $\vec{O}_i$, Eq.(7) can be reformulated as

$$\vec{O}_i = \left(\prod_{j=1}^{i-1} R(\theta_j) \right) R_i(\theta_i)\vec{L}_i = \tilde{R}_{i-1} R(\theta_i)\vec{L}_i \,, \tag{8}$$

then, θ_i can be solved from

$$R(\theta_i)\vec{L}_i = \left(\tilde{R}_{i-1}\right)^{-1} \vec{O}_i \,, \tag{9}$$

where we define $\tilde{R}_{i-1}$ is the function of $\tilde{\theta}_{i-1}$ which is an equivalent rotate angle, i.e.,

$$\tilde{R}_{i-1} = \prod_{j=1}^{i-1} R(\theta_j) = R(\tilde{\theta}_{i-1}), \tag{10}$$

and can be solved from

$$\vec{O}_{i-1} = R(\tilde{\theta}_{i-1})\vec{L}_{i-1} \,. \tag{11}$$

For example, assume

$$\vec{L}_i = [1 \quad 0]^T \text{, and } R(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}, \tag{12}$$

thus we can solve $\tilde{\theta}_{i-1}$ based on Eq.(11) from

$$\vec{O}_{i-1} = \begin{bmatrix} \cos\tilde{\theta}_{i-1} \\ \sin\tilde{\theta}_{i-1} \end{bmatrix}, \tag{13}$$

and obtain $\left(\tilde{R}_{i-1}\right)^{-1}$ for solving Eq.(9). Thus, θ_i can be obtained. Note that $\left(\tilde{R}_{i-1}\right)^{-1} = \left(\tilde{R}_{i-1}\right)^T$ because $\tilde{R}_{i-1}$ is the orthogonal matrix that results in $\left\|\vec{O}_{i-1}\right\| = \left\|\vec{L}_{i-1}\right\|$.

Consequentially, Eq.(9) to Eq.(11) can be used in order to obtain $\boldsymbol{\theta}^d$=[$\theta_1{}^d \ldots \theta_i{}^d \ldots \theta_n{}^d$]^T from { $\vec{O}_i{}^d$, i=1…n}. And, the desired joint velocity is computed by

$$\mathbf{z} = \boldsymbol{\theta}^d - \boldsymbol{\theta} \,. \tag{14}$$

Where, **z** is identical with that desired collision-free vector maps into the null-space projection in Eq.(3).

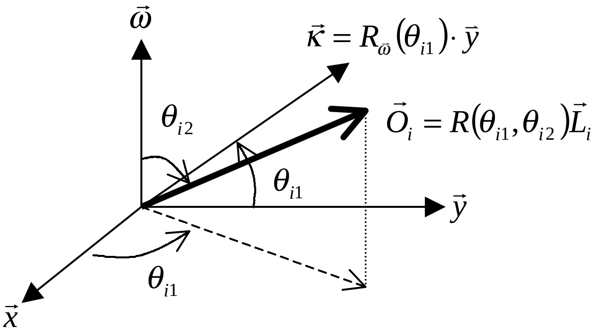
$\vec{\omega}$
$\vec{\kappa} = R_{\vec{\omega}}(\theta_{i1}) \cdot \vec{y}$
θ_{i2}
$\vec{O}_i = R(\theta_{i1}, \theta_{i2})\vec{L}_i$
θ_{i1}
$\vec{y}$
θ_{i1}
$\vec{x}$

Fig.2. The illustration for the rotation matrix $R(\theta_{i1}, \theta_{i2})$ in Eq.(16).

Difference of the posture generation between the planar and 3D manipulators is to define the rotation matrix. That is because the link to pose for arbitrary orientations in the 3D workspace needs at least two degrees of freedom. Therefore, we can define the rotation matrix R_i of the i^{th} joint is the function of $(\theta_{i1}, \theta_{i2})$ in which θ_{i1} and θ_{i2} are rotated about two mutually orthogonal axes orthogonal axes $\vec{\omega}$ and $\vec{\kappa}$, i.e., $\vec{\omega} \cdot \vec{\kappa} = 0$, where

$$\frac{\partial R(\theta_{i1}, \theta_{i2})}{\partial \theta_{i1}} \vec{O}_i = \vec{\omega} \times R(\theta_{i1}, \theta_{i2}) \vec{O}_i \text{, and}$$

$$\frac{\partial R(\theta_{i1}, \theta_{i2})}{\partial \theta_{i2}} \vec{O}_i = \vec{\kappa} \times R(\theta_{i1}, \theta_{i2}) \vec{O}_i \,. \tag{15}$$

Similar to the use of Eq.(9) to Eq.(11), the pair of angles $(\theta_{i1}, \theta_{i2})$ can be solved. For example,

$$\vec{L}_i = [0 \quad 0 \quad 1]^T \text{, and}$$

$$R(\theta_{i1}, \theta_{i2}) = \begin{bmatrix} \cos\theta_{i1} & -\sin\theta_{i1} & 0 \\ \sin\theta_{i1} & \cos\theta_{i1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_{i2} & 0 & \sin\theta_{i2} \\ 0 & 1 & 0 \\ -\sin\theta_{i2} & 0 & \cos\theta_{i2} \end{bmatrix}, \tag{16}$$

An illustration for the rotation matrix $R(\theta_{i1},\theta_{i2})$ is shown in Figure2, the vector $\vec{\varpi}$ is always perpendicular to the *xy*-plane and $\vec{\kappa}$ is changed with $R_{\vec{\varpi}}(\theta_{i1})\cdot \bar{y}$. Thus, we can obtain $(\tilde{\theta}_{i1},\tilde{\theta}_{i2})$ for $\tilde{R}_i$ from Eq.(16) by solving

$$\vec{O}_i = \begin{bmatrix} \cos\tilde{\theta}_{i1}\sin\tilde{\theta}_{i2} \\ \sin\tilde{\theta}_{i1}\sin\tilde{\theta}_{i2} \\ \cos\tilde{\theta}_{i2} \end{bmatrix}. \tag{17}$$

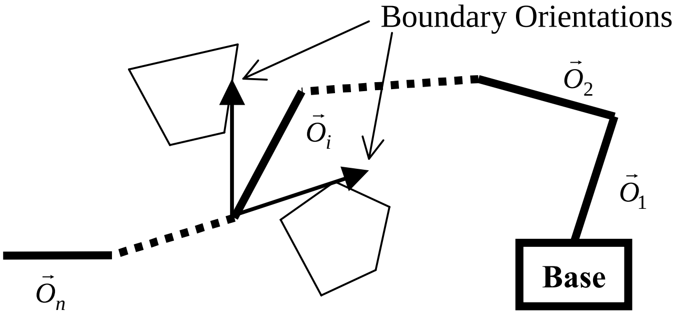

Fig.3. Boundary orientations for the *i*th link with the orientation $\vec{O}_i$.

4. Collision Avoidance

While the link is tracking an implicit path resulted from Eq.(6) in the workspace, the range of orientations may be bounded at the close obstacles. For example, Figure3 shows that the link $\vec{O}_i$ is bounded at the two polygonal obstacles. The proper posture for each of the links should keep away from these obstacles. Here, obstacles are represented as a set of convex polygons $\{P_j, j=1\ldots p\}$ for simplifying computation of collision detection. Moreover, each link $\vec{O}_i$ is shrunk as a line segment [H_{i+1}, H_i] when the set of polygons are extended in their sizes according to the real size of this link. Then, detection of collision is based on use of the minimum Euclidean distance d_{ij} between the line segment [$\mathrm{H}_{i+1}+\dot{\mathbf{x}}_{i+1}$, $\mathrm{H}_i+\dot{\mathbf{x}}_{i+1}$] (i.e. the link $\vec{O}_i$) and the polygon P_j. Generally, to keep the links away from $\{P_j, j=1\ldots k\}$ a specified safe distance d_s is expected. In other words, the orientation vector of link $\vec{O}_i$ should be driven toward a collision-free orientation $\vec{O}_i^{\,d}$ when $d_{ij} < d_\mathrm{s}$ for all P_j. For each link $\vec{O}_i$,

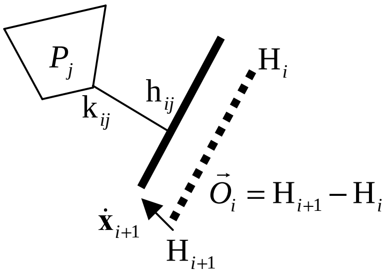

Fig.4. The minimum distance d_{ij} =$\|\mathrm{h}_{ij} - \mathrm{k}_{ij}\|$ between the convex polygon P_j and the link $\vec{O}_i$.

there might exist a subset $\{ P_j \mid d_{ij} < d_{\mathrm{s}} \}$. To generate the desired orientation vector $\vec{O}_i^{\,d}$ from $\vec{O}_i$, the method is suggested as follows.

Assume the minimum distance d_{ij} =$\|\mathrm{h}_{ij} - \mathrm{k}_{ij}\|$, where h_{ij} and k_{ij} are the closest points on the line segment $[\mathrm{H}_{i+1}+\dot{\mathbf{x}}_{i+1},\ \mathrm{H}_i+\dot{\mathbf{x}}_{i+1}]$ of link $\vec{O}_i$ and the convex polygonal obstacle P_j, respectively, such as that shown in Figure4. Due to the obstacles were represented convex, the vector $(\mathrm{h}_{ij} - \mathrm{k}_{ij})$ naturally goes toward a collision-free direction. The collision-free rotative direction can be obtained by differentiating the square minimum distance $d_{ij}{}^2(\tilde{\theta}_i)$ that is formulated as the function of $\tilde{\theta}_i$ (which has been defined in Eq.(11)), namely,

$$\frac{1}{d_{ij}\cdot\left\|\vec{L}_i\right\|}\cdot\frac{d\,d_{ij}{}^2(\tilde{\theta}_i)}{d\tilde{\theta}_i}, \tag{18}$$

where

$$d_{ij}{}^2(\tilde{\theta}_i)=\left\|-\frac{\left\|\mathrm{H}_{i+1}+\dot{\mathbf{x}}_{i+1}-\mathrm{h}_{ij}\right\|}{\left\|\vec{L}_i\right\|}R(\tilde{\theta}_i)\vec{L}_i+\mathrm{H}_{i+1}+\dot{\mathbf{x}}_{i+1}-\mathrm{k}_{ij}\right\|^2. \tag{19}$$

Noted that Eq.(18) is normalized with the multiplicator of $1\big/\left(d_{ij}\cdot\left\|\vec{L}_i\right\|\right)$ because $d_{ij}\left(\tilde{\theta}_i\right)$ is proportional to $\left\|\mathrm{H}_{i+1}+\dot{\mathbf{x}}_{i+1}-\mathrm{h}_{ij}\right\|$ and d_{ij}, where $0\le\left\|\mathrm{H}_{i+1}+\dot{\mathbf{x}}_{i+1}-\mathrm{h}_{ij}\right\|\le\left\|\vec{L}_i\right\|$. Then, we have the collision-free rotation angle

$$\Delta\tilde{\theta}_{ij}=\begin{cases}\dfrac{1}{d_{ij}\cdot\left\|\vec{L}_i\right\|}\cdot\dfrac{d\,d_{ij}{}^2\left(\tilde{\theta}_i\right)}{d\tilde{\theta}_i}\cdot\left(1-\left(\dfrac{d_{ij}}{d_s}\right)^2\right)\cdot\Delta\omega_i & ,\ d_{ij}<d_s\cdot, \\ 0 & ,\ d_{ij}\ge d_s.\end{cases}\tag{20}$$

where $\Delta\omega_i>0$ is a user-defined constant rotation angle that determines speed to escape from obstacles and should be smaller when the passageways between obstacles were narrow. One should be noted that $\Delta\tilde{\theta}_{ij}$ could be reduced more rapidly when d_{ij} was closer toward d_s, which results in effective collision avoidance when the link is closer to the obstacle.

To get a compromise among the set $\{\Delta\tilde{\theta}_{ij},\ j=1\ldots p\}$ for obtaining $\vec{O}_i{}^d$, $\{\Delta\tilde{\theta}_{ij},\ j=1\ldots p\}$ can be separated into two subsets $\{\Delta\tilde{\theta}_{ij}\mid\ \Delta\tilde{\theta}_{ij}>0\}$ and $\{\Delta\tilde{\theta}_{ij}\mid\ \Delta\tilde{\theta}_{ij}<0\}$ by the different rotation directions. Then, we can calculate a compromised collision-free rotation angle by

$$\Delta\tilde{\theta}_i=\max\left\{\Delta\tilde{\theta}_{ij}\mid\Delta\tilde{\theta}_{ij}>0\right\}+\min\left\{\Delta\tilde{\theta}_{ij}\mid\Delta\tilde{\theta}_{ij}<0\right\},\tag{21}$$

and finally we have

$$\vec{O}_i{}^d=R\left(\Delta\tilde{\theta}_i\right)\vec{O}_i.\tag{22}$$

In addition to obtain the $\Delta\tilde{\theta}_{ij}$ and $\Delta\tilde{\theta}_i$ in the 2D workspace, the pair of angles $\left(\Delta\tilde{\theta}_{i1j},\Delta\tilde{\theta}_{i2j}\right)$ that defined in Eq.(15) for representing the link $\vec{O}_i$ in the 3D workspace can similarly be calculated from

$$\Delta\tilde{\theta}_{igj}=\begin{cases}\dfrac{1}{d_{ij}}\cdot\dfrac{\partial\,d_{ij}{}^2\left(\tilde{\theta}_{i1},\tilde{\theta}_{i2}\right)}{\partial\tilde{\theta}_{ig}}\cdot\left(1-\left(\dfrac{d_{ij}}{d_s}\right)^2\right)\cdot\Delta\omega_i & ,\ d_{ij}<d_s\cdot, \\ 0 & ,\ d_{ij}\ge d_s.\end{cases}\tag{23}$$

where g=1,2. Similar to Eq.(21), we can compute the compromised rotation angles by

$$\Delta\tilde{\theta}_{ig} = \max\left\{\Delta\tilde{\theta}_{igj} \mid \Delta\tilde{\theta}_{igj} > 0\right\} + \min\left\{\Delta\tilde{\theta}_{igj} \mid \Delta\tilde{\theta}_{igj} < 0\right\}, \text{ for } g=1,2. \tag{24}$$

Finally, we have

$$\vec{O}_i^{\,d} = R\left(\Delta\tilde{\theta}_{i1}, \Delta\tilde{\theta}_{i2}\right)\vec{O}_i\,. \tag{25}$$

On the other hand, the joint limit can be concerned for bounding generating each of the orientation vectors $\vec{O}_i^{\,d}$. Based on the proposed principle, each $\vec{O}_i^{\,d}$ would be bounded on the $(i+1)^{th}$ joint limit. Thus, $\Delta\tilde{\theta}_i$ or $\left(\Delta\tilde{\theta}_{i1}, \Delta\tilde{\theta}_{i2}\right)$ might be pruned because of the joint limit, which may result in an orientation vector $\vec{O}_i^{\,d}$ that approximates to the really desired one as far as possible. For simplifying implementation of simulations in Section 5, the joint limits, however, would not be concerned.

5. Simulation Results

In this section, we have completed the simulations in the 2D Cartesian space in which the links were shrunk into line segment and obstacles were represented by convex polygons. In Figure5, the planar redundant manipulator with 11 DOFs was employed to perform the obstacle avoidance when the end-effector was tracking the path consisting of four line segments toward a goal inside the group of eight convex polygonal obstacles. Note that the polygons are cluttered closely, namely, passages of the path are narrow. Based on the use of posture generation principle and determination of link orientations, Figure5 shows the good performance of obstacle avoidance when the end-effector reached several positions of the path. Where, the ratio of lengths between the links is specified from the base to the end-effector as [3, 2.8, 2.4, 1.8, 1.8, 1.8, 1.6, 1.6, 1.4, 1.2, 1], and the parameters d_s=0.7 and $\Delta\omega_i$ =10°.

In addition to the collision avoidance in the static environment, the proposed technique can be applied to the dynamic environment in which obstacles may be moved. As the simulation in Figure6, the redundant manipulator with 9 DOFs could safely maneuver between a static polygon and a dynamic polygon moving from the right to left sides of the workspace, when the end-effector was tracking part of the edges of the static polygons. Where, the ratio of lengths between the links is specified as [2, 1.8, 1.6, 1.4, 1.2, 1, 0.9, 0.8, 0.7], and d_s=0.3 and $\Delta\omega_i$ =10°. The performance of simulation result was good as expected.

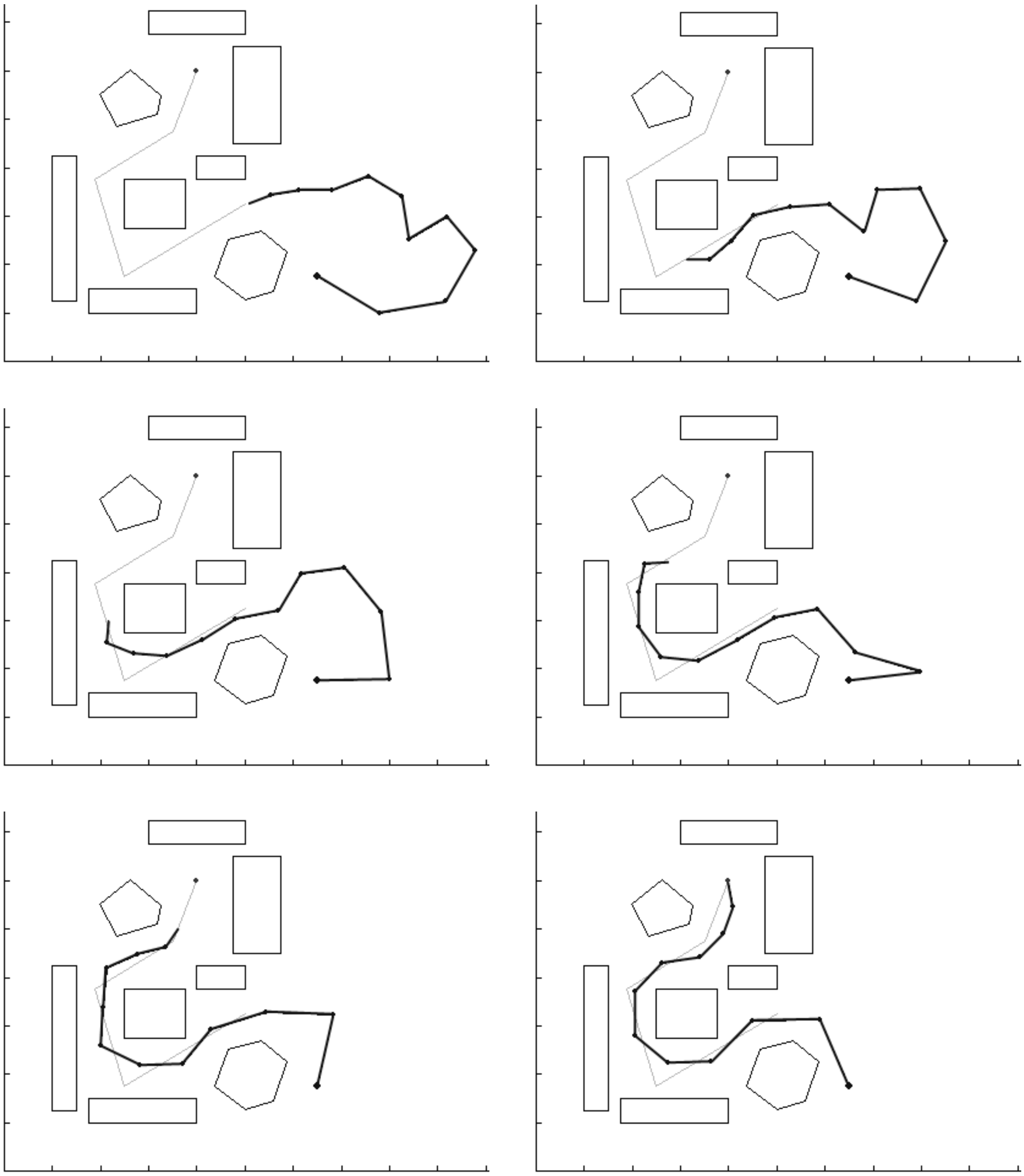

Fig.5. The good performance of obstacle avoidance when the end-effector reached several positions of the specified path.

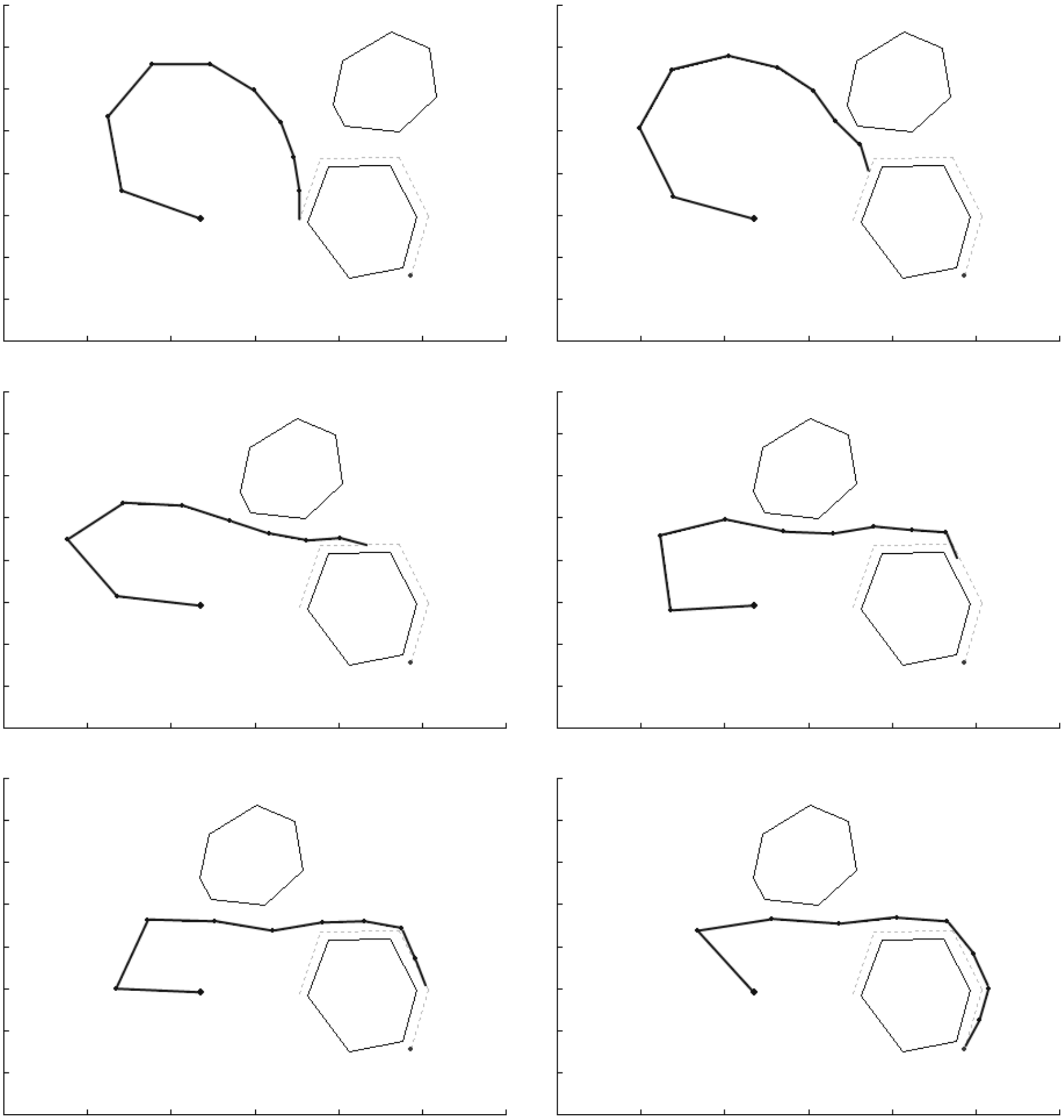

Fig.6. The good performance of obstacle avoidance when the environment is dynamic.

6. Conclusions

In this report, we proposed a principle useful for on-line globally planning postures of the redundant manipulator, which can be incorporated with the technique of pseudoinverse control. The principle suggests backward determining the link orientations from the end-effector to the base because of the fact indicated in Eq.(6). By the principle, the problem of posture generation for the redundant manipulator can be transformed from searching a joint vector in the joint space into determining a set of orientation vectors in the workspace. Therefore, the proposed technique of collision avoidance can be on-line applied to both static and dynamic environments.

Based on use of the principle, a method is suggested to determine the collision-free orientation vectors of links while the obstacles are represented as convex polygons in the workspace. The method can obtain the compromised rotative angles for generating the collision-free orientation vectors that allows the links to safely move among the closely cluttered obstacles. The proposed collision avoidance technique had been simulated in the 2D static and dynamic environments, respectively. In future work, we will try to apply the principle to the other issues of redundant manipulator.

REFERENCES

1. J. Barraquand and J. C. Latombe, “Robot motion planning: A distributed representation approach,” Int. J. of Robotics Research, vol. 10, no. 6, pp.628-649, 1991.
2. J. C. Latombe, “Robot motion planning,” Kluwer Academic Publishers, 1991.
3. T. Lozano-Pérez, “Spatial planning: A configuration space approach,” IEEE Trans. On Computers, vol. 32, no. 2, pp. 108-120, 1983.
4. J. A. Kuo and D. J. Sanger, “Task planning for serial redundant manipulators,” Robotica, vol. 15, pp. 75-83, 1997.
5. T. C. Liang and J. S. Liu, “An improved trajectory planner for redundant manipulators in constrained workspace,” J. of Robotic Systems, vol. 16, no. 6, pp. 339-351, 1999.
6. E. S. Conkur and R. Buckingham, “Manoeuvring highly redundant manipulators,” Robotica, vol. 15, pp. 435-447, 1997.
7. 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, vol. 13, no. 3, pp. 163-176, 1996.

8. M. T. H. Beheshti and A. K. Tehranl, "Obstacle avoidance for kinematically redundant robots using an adaptive fuzzy logic algorithm," In Proc. the American Control Conf., San Diego, California, June, 1999, pp. 1371-1375.
9. A. McLean, and S. Cameron, "The virtual springs method: path planning and collision avoidance for redundant manipulator," Int. J. of Robotics Research, vol. 15, no. 4, pp.300-319, 1996.
10. K. Glass, R. Colbaugh, D. Lim, H. Seraji, "On-line collision avoidance for redundant manipulator," In Proc. IEEE Int. Conf. Robot. and Automat., 1993, pp. 36-43.
11. K. Glass, R. Colbaugh, D. Lim, H. Seraji, "Real-time collision avoidance for redundant manipulators," IEEE Trans. On Robot. and Automat., vol. 11, no. 3, pp. 448-457, June, 1995.
12. W. S. Newman, "Automatic obstacle avoidance at high speeds via reflex control," In Proc. IEEE Int. Conf. Robot. and Automat., 1989, pp. 1104-1109.
13. T. S. Wikman, W. S. Newman, "A fast, on-line collision avoidance method for a kinematically redundant manipulator based on reflex control," In Proc. IEEE Int. Conf. Robot. and Automat., 1992, pp. 261-266.
14. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," Int. J. of Robotics Research, vol. 5, no. 1, pp.90-98, 1986.
15. Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based on redundant redundancy control of robot manipulators," Int. J. of Robotics Research, vol. 6, no. 2, pp.3-15, 1987.
16. S. I. Choi and B. K. Kim, "Obstacle avoidance control for redundant manipulators using collidability measure," Robotica, vol. 18, pp. 143-151, 2000.
17. S. I. Choi and B. K. Kim, "Obstacle avoidance for redundant manipulators using directional-collidability/temporal-collidability measure," Journal of Intelligent and Robotic Systems, vol. 28, no. 3, pp. 213-229, 2000.
18. Z. Y. Guo and T. C. Hsia, "Joint trajectory generation for redundant robots in an environment with obstacles," In Proc. IEEE Int. Conf. Robot. and Automat., 1990, pp. 157-162.
19. N. Rahmanian-Shahri and I. Troch, "Collision-avoidance control for redundant articulated robots," Robotica, vol. 13, pp. 159-168, 1995.
20. N. Rahmanian-Shahri and I. Troch, "A new on-line method to avoid collisions with links of redundant articulated robots," Robotica, vol. 14, pp. 611-619, 1996.
21. A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," Int. J. of Robotics Research, vol. 4, no. 3, pp.109-117, 1985.
22. W. J. Cho, D. S. Kwon, "A sensor-based obstacle avoidance for a redundant manipulator using a velocity potential function," In Proc. IEEE Int. workshop on Robot and Human communication, 1996, pp. 306-310.
23. J. Wunderlich and C. Boncelet, "Local optimization of redundant manipulator kinematics within constrained workspaces," In Proc. IEEE Int. Conf. Robot. and Automat., 1996, pp. 127-132.
24. H. Seraji, "Configuration control of redundant manipulators: theory and implementation,"

IEEE Trans. On Robot. and Automat., vol. 5, no. 4, pp. 472-490, August, 1989.

25. H. Seraji and R. Colbaugh, “Improved configuration control for redundant robots,” J. of Robot. Syst., vol. 7, no. 6, pp. 897-928, 1990.
26. C. Klein and C. Huang, “Review of pseudoinverse control for use with kinematically redundant manipulators,” IEEE Trans. On Syst., Man and Cyber., vol. 13, no. 3, pp. 245-250, 1983.
27. S. Chiaverini, “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulator,” IEEE Trans. On Robot. and Automat., vol. 13, no. 3, pp. 398-410, June, 1997.
28. R. G. Roberts and A. A. Maciejewski, “Singularities, stable surface, and the repeatable behavior of kinematically redundant manipulators,” Int. J. of Robotics Research, vol. 13, no. 1, pp.70-81, 1994.
29. D. Li, A. A. Goldenberg, and J. W. Zu, “Peak torque reduction with redundant manipulators,” In Proc. IEEE Int. Conf. Robot. and Automat., 1996, pp. 1775-1780.
30. J. M. Hollerbach and K. I. C. Suh, “Redundancy resolution of manipulators through torque optimization,” IEEE J. of Robot. and Automat., vol. 3, pp. 308-316, 1984.
31. P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot, “Evaluation of force capabilities for redundant manipulators,” In Proc. IEEE Int. Conf. Robot. and Automat., 1996, pp. 3520-3525.
32. B. Hu, C. L. Teo, and H. P. Lee, “Local optimization of weighted joint torques for redundant robotic manipulators,” IEEE Trans. On Robot. and Automat., vol. 11, no. 3, pp. 422-425, June, 1995.
33. R. G. Roberts and A. A. Maciejewski, “A local measure of fault tolerance for kinematically redundant manipulators,” IEEE Trans. On Robot. and Automat., vol. 12, no. 4, pp. 543-552, Aug., 1996.
34. J. D. English and A. A. Maciejewski, “Fault tolerance for kinematically redundant manipulators anticipating free-swinging joint failures,” In Proc. IEEE Int. Conf. Robot. and Automat., 1996, pp. 460-467.