

## Motion Planning Algorithms of Redundant Manipulators Based on Self-motion Manifolds

YAO Yufeng\*, ZHAO Jianwen, and HUANG Bo

*State Key Laboratory of Robotics Technique and System, Harbin Institute of Technology, Harbin 150081, China*

Received March 3, 2009; revised December 18, 2009; accepted December 22, 2009; published electronically December 24, 2009

**Abstract:** The current motion planning approaches for redundant manipulators mainly includes two categories: improved gradient-projection method and some other efficiency numerical methods. The former is excessively sensitive to parameters, which makes adjustment difficult; and the latter treats the motion planning as general task by ignoring the particularity, which has good universal property but reduces the solving speed for on-line real-time planning. In this paper, a novel stepwise solution based on self-motion manifold is proposed for motion planning of redundant manipulators, namely, the chief tasks and secondary tasks are implemented step by step. Firstly, the posture tracking of end-effector is achieved accurately by employing the non-redundant joint. Secondly, the end-effector is set to keep stationary. Finally, self-motion of manipulator is realized via additional work on the gradient of redundant joint displacement. To verify this solution, experiments of round obstacle avoiding are carried out via the planar 3 degree-of-freedom manipulator. And the experimental results indicate that this motion planning algorithm can effectively achieve obstacle avoiding and posture tracking of the end-effector. Compared with traditional gradient projection method, this approach can accelerate the problem-solving process, and is more applicable to obstacle avoiding and other additional work in displacement level.

**Key words:** redundant manipulators, self-motion, motion planning, inverse kinematics

### 1 Introduction

The purpose of the redundant manipulator is assumed to optimize manipulator motion performance with a redundant degree-of-freedom (DOF), which is normally referred to as “motion planning”. The manipulator motion planning encompasses the planning of relative position, velocity and acceleration. Planning of velocity and acceleration should favorably allow for the probability of end-effector satisfying the prescribed pose in real time, as a result, the motion planning on the displacement level turns out to be essential for that of redundant manipulator. Unfortunately, the optimization algorithm for inverse kinematics on displacement level is rarely considered. However, in the inverse arithmetic of gradient projection on velocity level, the gradient of performance function is creatively integrated into a velocity inverse solution that significantly facilitates the search work for inverse solution optimization, which was reasonably adapted extensively. Its expression could be described as follows:

$$\dot{\theta} = J^+ \dot{x} + k(I - J^+ J) \nabla H(\theta),$$

where  $J^+ = J^T(JJ^T)^{-1}$ ,  $k$  is the magnification coefficient;

$\nabla H$  is the gradient for the performance criteria function  $H(\theta)$  of the additional task.

Although the calculating process of gradient projection is simple, and its geometric significance is explicit, it has following drawbacks:

(1) The heavy computation for pseudo-inverse  $J^+$  makes the real-time computation rather difficult.

(2) The error in solving inverse allows for only an imprecise optimization solution.

(3) The tracing of end-effector displacement could be realized only by integral velocity, thus will inevitably result in error accumulation.

(4) A significant difference between a special solution and an homogeneous solution may occur if the magnification coefficient  $k$  of real-value scalar goes to the two extremes, and in the context of velocity resolution method, the element of magnitude order being too small will be disregarded, which will eventually lead to the failure of the gradient projection method(GPM).

To this end, scholars have offered several proposals for selection of  $k^{[1]}$ , among which the real-time correction scheme for  $k$  proposed by WU, et al<sup>[2]</sup>, proves to achieve better result,  $k$  is defined as follows:

$$k = \psi \frac{\|J^+ \dot{x}\|}{\|(I - J^+ J) \nabla H\|}, \quad (1)$$

where  $\psi$  is adjustment coefficient, and its value may

\* Corresponding author. E-mail: hitwhyf@hitwh.edu.cn  
This project is supported by National Hi-tech Research and Development Program of China (863 Program, Grant No. 2005AA404291)

range within  $(-0.2, -1.0)$ . For computation within this scope, the inverse solution for redundant manipulator remains stable in normal cases; however, the optimization result will be discounted significantly at the same time.

In recent years, scholars have been dedicated to perfecting the gradient projection all the way, the example proposed in Ref. [3]—the secondary calculation method—is a case in point, in which the GPM is used to secure the imprecise solution first, followed by the secondary calculation in compliance with the feature of system configuration, until finally producing a set of precise optimization solutions. However, this method is problematically incomplete since great fluctuation on joint velocity cannot be avoided even if it is possible to overcome the first three drawbacks of the gradient projection method. Ref. [4] presents a modified GPM, in which the pose change of end-effector generates reference velocity in iterative process, thus guarantees that the end-effector will converge to the desired position and pose based on the precision prescribed by subscribers.

The GPM, in nature, is no more than an optimization method on velocity level, which could be applied to motion planning of velocity level; however, it is not the best for tasks of precise trajectory tracking, obstacle avoidance etc. Many scholars attempted a number of numerical methods to keep the end-effector in high pose precision<sup>[5–6]</sup> while ensuring that the manipulator performs the additional task in the meantime. Refs. [4–6] apparently discard the drawback of end pose error accumulation existed in the conventional gradient projection though, the gap between tracing precision and computation time could hardly be bridged. Still the methods described in Refs. [4–6] are far from satisfying when high displacement tracking precision is required for the manipulators.

In view of the drawbacks displayed in the mentioned methods, scholars begin to attempt new methods for motion planning from the perspective of self-motion—the peculiar motion to redundant manipulators. In pursuit of algorithm universality, the ordinary spatial redundant robot finds it hard to secure an analytical solution for the self-motion manifold, for this reason, the scholars who engage in inverse solution optimization searching from the perspective of self-motion more focus on controlling method<sup>[7–8]</sup> or retaining the algorithm on velocity level<sup>[9]</sup>. Subsequently the performance of an additional task on the position level remains weak.

For the redundant manipulators used in engineering projects, the geometric relation among joint axis is simple, this implies that the analytical solution of self-motion manifolds are easily reachable, so this supports the feasibility of performing inverse solution optimization searching based on self-motion manifold. In this paper, the inverse solution optimization searching on the displacement level from the perspective of self-motion manifold will be directly accomplished, with the intention of developing a

more efficient motion planning method for additional task at displacement level.

## 2 Derivation of motion planning algorithm

For those manipulators with simple geometric relations of joint axes, the self-motion manifold could be expressed by the displacement variables of a certain joints. In this paper, these joints were defined as redundant ones, and the column vector which is composed of each individual redundant joint displacement was defined as self-motion variable  $\theta_s$ . If  $r$  is redundant DOF, then  $\theta_s$  is  $r$ -dimensional column vector. Thus its forward kinematics could be represented via the self-motion variable  $\theta_s$  as

$$T_m = f(\theta, \theta_s),$$

where  $T_m$  is the pose matrix of end-effector,  $\theta$  is the joint angle vector of the manipulator.

Without lost of generality, the movement of end-effector from  $T_m$  to  $T'_m$  under the constraint of additional task could be fulfilled in two steps based on the kinematics equivalent concept: (1) lock redundant joint in a certain displacement, the pose matrix  $T'_m$  of end-effector is left to the motion of the remaining joints; (2) open redundant joint, the self-motion of the mechanism occurs in  $T'_m$ , hence to perform the additional task.

Let the self-motion manifold of mechanism in  $T'_m$  be  $C_{SM}$ , when the manipulator has  $n$  arm configuration signs,  $C_{SM}$  could be represented by  $2^n$  curves (redundant DOF beyond 1) or curved surface (redundant DOF below 1) in joint configuration space, with every curve or curved surface corresponding to a range of different arm configuration sign combination. When the manipulator is required to complete continuous task tracking, the subscriber could select one continuous curve or curved surface  $C_{SMi}$  from  $C_{SM}$  to carry out the additional task in the meantime.  $C_{SMi}$  could be expressed as parameter equation of self-motion variable  $\theta_s$ :

$$\theta = G_i(\theta_s).$$

In a general case, the property index  $H_v$  of additional task could be denoted by function  $H(\theta)$  of joint angle vector  $\theta$ , based on the above formula,  $H_v$  could be expressed as the function of  $\theta_s$ :

$$H_v = H[G_i(\theta_s)].$$

Inspired by the steepest descent method, it is arrived that  $\theta_s$  should change in the direction of gradient  $\nabla H_v$  as  $H_v$  for  $\theta_s$  in order to obtain the speediest decreasing velocity, when  $H(\theta)$  conforms to the minimal performance function law, the difference form  $\theta_s$  could be put as follows:

$$\begin{cases} \theta_s = \theta_{s0} - kH_v \frac{\nabla H_v}{|\nabla H_v|}, & |\nabla H_v| \neq 0, \\ \theta_s = \theta_{s0}, & |\nabla H_v| = 0, \end{cases} \quad (2)$$

where  $k$  is scale factor. We bring  $H_v$  into the coefficient in order to keep it in a downward direction corresponding to the index law. When the derivative of  $H_v$  corresponding to  $\theta_s$  is zero,  $H_v$  remains to be a constant below threshold value, implying that the mechanism is in an optimal state, with no need for additional task. Thus the geometric significance of Eq. (3) could be stated as follows: to locate the changing direction of  $\theta_s$  on the curve self-motion manifold, to bring down  $H_v$ .

When redundant DOF of the manipulator exceeds 1,  $H_v$  is the multivariate function for  $\theta_s$ , the solving formula for  $\theta_s$  should be modified as follows:

$$\begin{cases} \theta_s = \theta_{s0} - kH_v \operatorname{sgn}\left(\frac{dH_v}{d\theta_s}\right), & \left|\frac{dH_v}{d\theta_s}\right| \neq 0, \\ \theta_s = \theta_{s0}, & \frac{dH_v}{d\theta_s} = 0, \end{cases} \quad (3)$$

where  $\operatorname{sgn}(\bullet)$  is the symbol function.

The above mentioned algorithm could be summarized as follows:

(1) First to effect the precise position and pose tracking of the end-effector by redundant joints; then adjust the redundant joints by the gradient of redundant joint displacement through the additional task.

(2) In order to perform the additional task by employing the self-motion of the manipulator.

(3) Its geometric significance is to locate the changing direction for  $\theta_s$  on the self-motion manifold, in order to bring down  $H_v$  value to the minimum in the maximum speed. Eq. (3) can be regarded as the special form of Eq. (2).

The algorithm mentioned above can be illustrated in program chart displayed in Fig. 1, where  $S_{\text{sum}}$  is the number of sampling points.

### 3 Algorithm Application

In order to prove the effectiveness of the motion planning algorithm proposed in section 1, we used the simplest planar 3-DOF redundant manipulator to demonstrate how the collision avoidance simulation task is implemented. Its mechanism diagram is shown in Fig. 2. Initial position of the mechanism stays on the positive half axis of axis  $y$ , the rotation angles of the three joints are marked as  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ , respectively. The angles range within  $(-\pi, \pi]$ , and the values are positive when rotating clockwise around axis  $z$ .

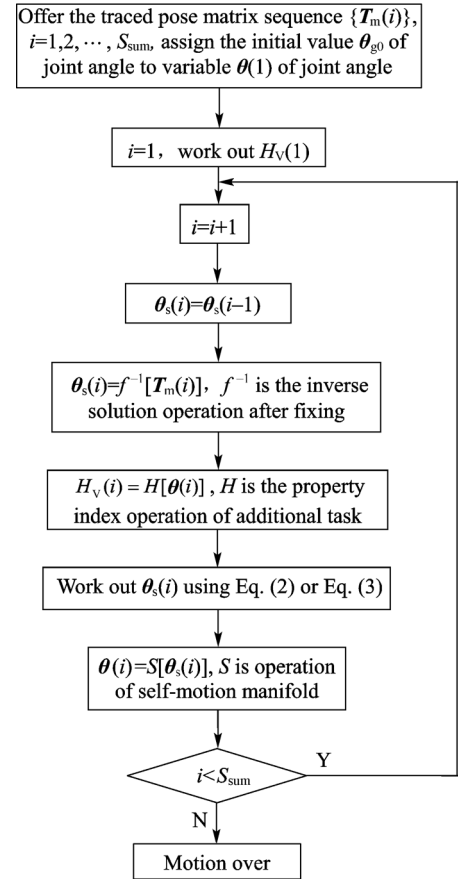


Fig. 1. Program chart for motion planning algorithm

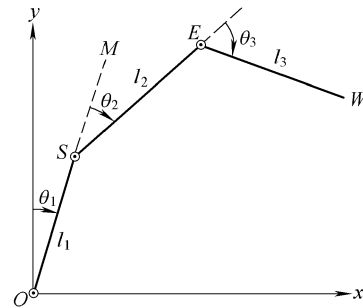


Fig. 2. Mechanism diagram of 3-DOF redundant manipulator

#### 3.1 Self-motion manifold of the mechanism

The manipulator has one redundant DOF when disregarding of the posture of the tip executing apparatus of the robot. It can be seen that there are four arm configuration signs in the mechanism, as indicated in Fig. 3.

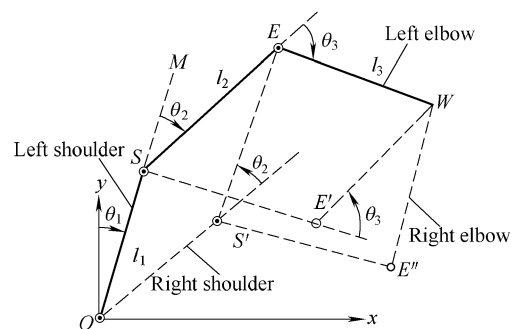


Fig. 3. Four kinds of arm configuration

For the sake of simplifying the motion analysis, we take joint 1 as redundancy joint,  $\theta_1$  as the variable of self-motion, when  $\theta_1$  value is specified, we get only two kinds of arm configuration signs in the mechanism. Define arm configuration signs:

$$e = \begin{cases} +1, & \text{left elbow,} \\ -1, & \text{right elbow.} \end{cases}$$

Suppose the link lengths of the manipulator are  $l_1$ ,  $l_2$ , and  $l_3$ , respectively, and the coordinate of point  $W$  is  $(x_W, y_W)$ , then we can conclude the following:

$$\overrightarrow{OW} = (x_W \quad y_W \quad 0)^T,$$

$$\overrightarrow{OS} = (l_1 \sin \theta_1 \quad l_1 \cos \theta_1 \quad 0)^T,$$

$$\overrightarrow{SW} = (x_W - l_1 \sin \theta_1 \quad y_W - l_1 \cos \theta_1 \quad 0)^T.$$

Let  $|\overrightarrow{SW}| = a$ , thus the following equation holds:

$$\theta_3 = e \cdot \left( \pi - \arccos \frac{l_2^2 + l_3^2 - a^2}{2l_2 l_3} \right).$$

Let's define

$$\theta_{2m} = \mathbf{u}_{\overrightarrow{SW}} \times \mathbf{u}_{\overrightarrow{OS}} \cdot \mathbf{z} \cdot \arccos(\mathbf{u}_{\overrightarrow{OS}} \cdot \mathbf{u}_{\overrightarrow{SW}}) - e \cdot \angle WSE,$$

where  $\mathbf{u}_{\overrightarrow{OS}} = \frac{\overrightarrow{OS}}{l_1}$ ,

$$\mathbf{u}_{\overrightarrow{SW}} = \frac{\overrightarrow{SW}}{a},$$

$$\mathbf{z} = (0 \quad 0 \quad 1)^T,$$

$$\angle WSE = \arccos \frac{a^2 + l_2^2 - l_3^2}{2al_2}.$$

In case  $\theta_{2m} \in (-\pi, \pi]$ , we take  $\theta_2 = \theta_{2m}$ ; in case  $\theta_{2m} \notin (-\pi, \pi]$ , we take  $\theta_2 = \theta_{2m} - \text{sign}(\theta_{2m}) \cdot 2\pi$ ,  $\text{sgn}(\cdot)$  is the symbol function.

The above-mentioned solving process proves that both  $\theta_3$  and  $\theta_2$  can be expressed in  $\theta_1$ . Let any point on the self-motion manifold be  $\boldsymbol{\theta} = (\theta_1 \quad \theta_2 \quad \theta_3)^T$ , we can use the solution to express  $\boldsymbol{\theta}$  as the function of  $\overrightarrow{OW}$  and  $\theta_1$ :

$$\boldsymbol{\theta} = f(\overrightarrow{OW}, \theta_1), \quad \overrightarrow{OW} \in U, \quad \theta_1 \in (\theta_{1L}, \theta_{1R}]. \quad (4)$$

Where  $U$  is motion region of the point  $W$ , while  $\theta_{1L}$ ,  $\theta_{1R}$  are left and right limit of the joint 1 rotation angle respectively. Eq. (4) is the function expression of the self-motion manifold. Let's take the arm configuration sign  $e = \pm 1$ , we can get the two curves of the self-motion manifold corresponding to  $\overrightarrow{OW}$  in the joint configuration space.

### 3.2 Obstacle avoidance motion planning based on self-motion manifold

Now the case of every link avoiding cycle obstacle is cited to illustrate the specific solution procedure.

#### 3.2.1 Establishment of performance index function for obstacle avoidance

As indicated in Fig. 4, select arm configuration sign  $e = 1$ , to establish performance index function for collision avoidance when obstacle locating on each link in clockwise.

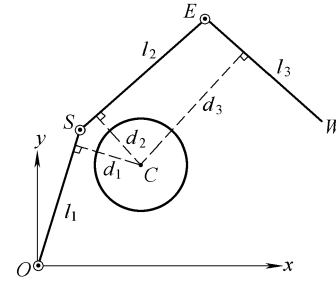


Fig. 4. Relative position diagram between each link and the obstacle

Set the distances from point  $C$ , the center of obstacle, to each bar are  $d_1$ ,  $d_2$ , and  $d_3$ , respectively, set the radius of round obstacle is  $r$ , and define:

$$d_{\min} = \min(d_1, d_2, d_3).$$

The conclusion is obvious, when  $d_{\min} > r$ , each link can be free from collision with obstacle. The performance function of obstacle avoidance is thus obtainable in compliance with minimal performance function rule and artificial potential field theory:

$$H_v = \begin{cases} k_p (d_{\min} - \lambda)^2, & d_{\min} \leq \lambda, \\ 0, & d_{\min} > \lambda, \end{cases} \quad (5)$$

where  $k_p$  is proportional coefficient,  $\lambda$  is threshold value of obstacle avoidance distance.

From the above function definition formula, we find that, when  $d_{\min}$  is equal to  $\lambda$ , the function obtains the minimal value, also the potential energy minimal point. By this way, the obstacle avoidance problem of mechanism link becomes optimization of minimal potential energy.

#### 3.2.2 Simulation process of obstacle avoidance

Suppose that  $l_1$ ,  $l_2$ ,  $l_3$  are 100 mm respectively and the initial angle  $\theta_1$  is  $\pi/6$ ,  $\theta_2$  is  $\pi/6$ ,  $\theta_3$  is  $\pi/2$ , then the initial coordinate (85, 60) of point  $W$  can be derived. Suppose that the diameter of the obstacle is 80 mm and its circle centre coordinate is (85, 60), besides, the desired track of point  $W$  is line segment  $WW'$ , the coordinate of  $W'$  is (146.6, 10), thus the initial state of simulation can be described as Fig. 5.

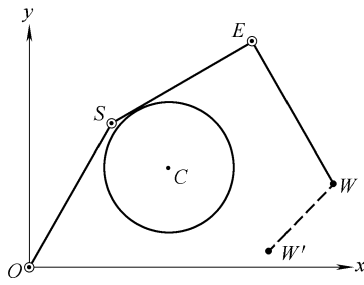


Fig. 5. Initial configuration of mechanism

Based on Eq. (5), set  $k_p=1$  and  $\lambda=45$ , equally divide line segment  $WW'$  into 10 parts, and then trace the above 11 interpolation points so as to implement the additional obstacle avoidance task by using the algorithm shown in Fig. 1. The configuration of each link in motion is shown in Fig. 6, and it shows that the mechanism complete the main task (tracking the reference point of terminal position) along with the additional task (avoiding obstacle  $C$ ) simultaneously.

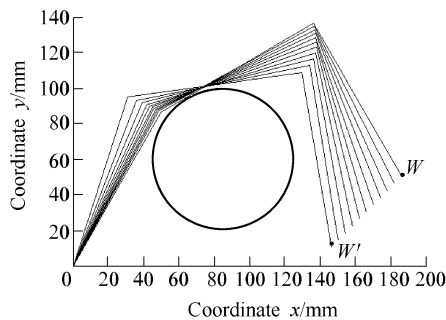


Fig. 6. Configuration of each link in the first simulation

Change of each joint angle in motion as indicated in Fig. 7, shows that the changes of joint angle displacement tend to be mild. Although the inverse solution is worked out on the position level, the joint velocity experiences no sharp fluctuation.

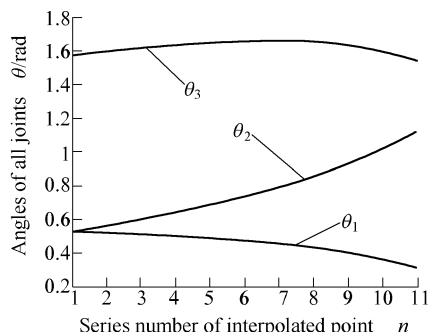


Fig. 7. Change of each joint angle in motion

The minimal distance  $d_{\min}$  from each link to circle center  $C$  and the change of performance index  $H_v$  are shown in Fig. 8. It is noticeable from Fig. 8 that  $d_{\min}$  remains larger than the circle radius, which means collision-free between link and the obstacle. The

performance index keeps on declining, meaning that obstacle avoidance algorithm works consistently over time; the performance index is in direct proportion to decline velocity, allowing for fast decline velocity when the performance index grows, and vice versa, which may ensure that with the success of obstacle avoidance, redundancy resources could be released quickly, thus enabling the implementation of other additional tasks.

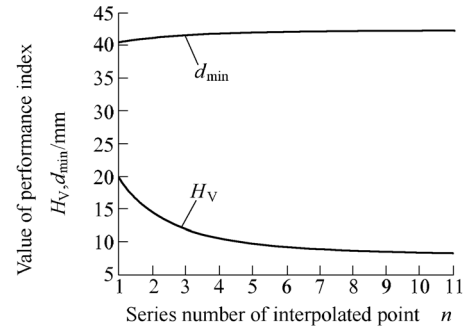


Fig. 8. Change of the minimal distance and performance index in motion

### 3.2.3 Algorithm application analysis

Some drawbacks still exist in joint velocity optimization as the above algorithm is produced at the displacement level, yet the velocity fluctuation could be constrained considerably by adjusting the relevant parameters. Let's illustrate this in greater detail by simulation.

Take initial joint angle  $\theta_1 = \pi/4$ ,  $\theta_2 = \pi/12$ ,  $\theta_3 = \pi/2$ , the initial coordinate of point  $W$  now is  $(207.27, 34.12)$ . Suppose the obstacle diameter is 60 mm that its circle centre coordinate is  $(100, 45)$ . The expected displacement of point  $W$  is line segment  $WW'$ , the coordinate of  $W'$  is  $(157.27, -15.88)$ , and other parameters are the same as in simulation 1. Now divide line segment equally  $WW'$  into 20 parts, trace the above 21 interpolation points, simulation time 2 s. Fig. 9 shows the configuration of each link in motion, it is visible from the diagram that when the link is in proximity to the obstacle, the redundancy joint movement speeds up, and slows down as the member depart from the obstacle; a second approach to the obstacle, the redundancy joint picks up speed again.

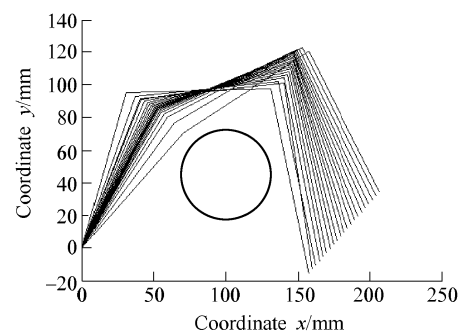


Fig. 9. Configuration of each link in the second simulation

Fig. 10 displays changing pattern of each joint velocity before adjustment. In the initial moment, the link gets quite close to the obstacle, allowing for tremendous effect on the mechanism from the additional task, which explains the great fluctuation of each joint velocity explicitly.

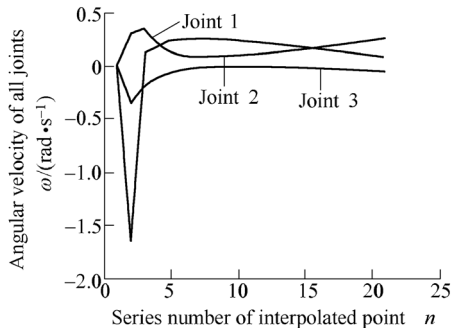


Fig. 10. Changing pattern of each joint velocity before adjustment

In order to reduce the fluctuation of joint angle velocity, we choose to diminish the impact on mechanism motion from the additional task, as a big margin still exists in  $\lambda$  value selection, the threshold value  $\lambda$  of obstacle avoidance distance can be brought down to  $\lambda = 40$ , now each joint angle velocity fluctuates slightly, as shown in Fig. 11.

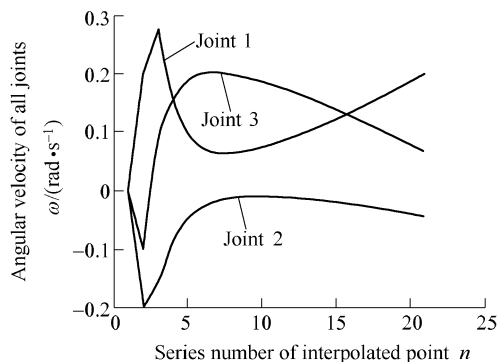


Fig. 11. Changing pattern of each joint velocity after adjustment

#### 4 Algorithm Application Example

In order to further confirm the workability of the said algorithm in engineering application, let's take the window operation of 8-DOF mobile manipulator as an example. The mobile manipulator has two parts, the mobile platform and mechanical arms<sup>[10]</sup>, its simplified model is demonstrated as shown in Fig. 12. The joint configuration of the mechanical arms is identical to the robot configuration of the traditional PUMA560, the 6 joints are roll-pitch-pitch-roll-pitch-roll.

In most cases the mobile manipulator would conduct its window operation vertical to the wall surface motion, which makes it critical to analyze the window operation of the robot inside the wall vertical face. As shown in Fig. 13,

we simplify the mobile platform into the rectangle  $HIJK$ , and the main arm and lower arm into line segments  $PE$  and  $EW$ , the pitch motion of the main arm could be displayed by the line segment  $PE$  rotating around the point  $P$ ; while that of the lower arm by the line segment  $EW$  rotating around point  $E$ . Line segment  $PK$  is consolidated to the mobile platform in order to measure the installation height of the mechanical arm.

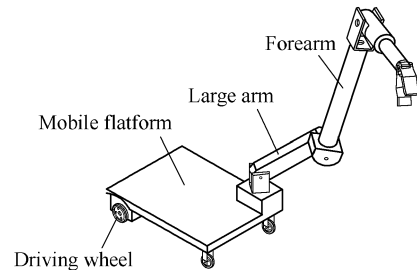


Fig. 12. Simplified model of 8-DOF mobile manipulator

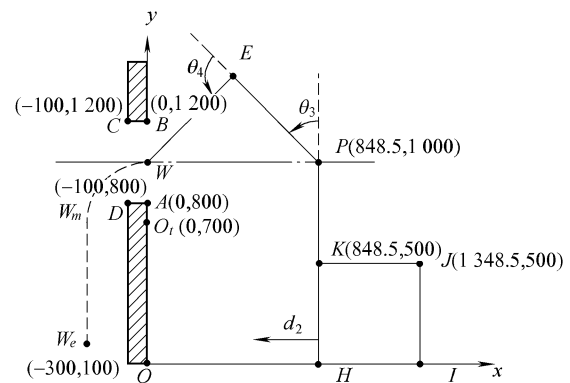


Fig. 13. Schematic drawing for primary configuration and tracking trajectory of the mobile manipulator

Point  $W$  is set as reference point of the end-effector, and this point is supposed to get through the window  $ABCD$  on the wall, and then reach the target point  $W_e$  along the curved line  $WW_mW_e$ , where  $WW_m$  is a quarter circular arc with circle center of point  $O_t$  and radius of 300 mm; and  $W_mW_e$  is a plumb line segment.

To build a plane coordinate system  $O_{xy}$  at the outer point  $O$  of the wall, the initial configuration and the coordinate of every points of the robot are shown in Fig. 13, the initial joint displacement is  $d_2 = 0$ ,  $\theta_3 = \pi/4$ , and  $\theta_4 = \pi/4$ .

Let the minimal distance from points  $A$ ,  $B$ ,  $C$ , and  $D$  to the line segment  $WE$  be  $D_1$ ,  $D_2$ ,  $D_3$ , and  $D_4$ , the minimal distance to the line segment  $EP$  be  $D_5$ ,  $D_6$ ,  $D_7$ , and  $D_8$ , respectively. It is clear from the mechanism movement feature that the line segment  $HP$  and the wall surface  $OA$  consistently parallel, let the spacing between them be  $D_9$ , we get the following equation  $D_9 = 848.5 - d_2$ . The problem-solving process for  $D_1 - D_8$  is slightly more complicated, however, the solving method follows the same pattern, let's take  $D_1$  as an example to illustrate the simplified solving method.

When  $\overline{WA} \cdot \overline{WE} < 0$ , the foot drop from point  $A$  to the

line  $WE$  lies on the extended line of the line segment  $EW$ , thus  $D_1 = |\overline{WA}|$ ; In the same way, when  $\overline{EA} \cdot \overline{EW} < 0$ , the foot drop from point  $A$  to the line  $WE$  lies on the extended line of the line segment  $WE$ , thus  $D_1 = |\overline{EA}|$ ; when  $\overline{WA} \cdot \overline{WE} \geq 0$  and  $\overline{EA} \cdot \overline{EW} \geq 0$ ,  $D_1 = |\overline{WA} \times \overline{WE}| / |\overline{WE}|$ .

Let's take the collision avoidance performance index function in Eq. (5), let  $d_{\min} = \min(D_1, D_2, \dots, D_8)$ ,  $k_p = 5 \times 10^{-4}$ ,  $\lambda = 400$  mm.

Now let's conduct the simulation of the mechanical arm performing the window operation using MATLAB software with the motion planning algorithm supplied in Fig. 1, take  $k = 0.6$  in Eq. (5), let  $|\overline{PE}| = |\overline{EW}| = 600$  mm, thus the configuration changing pattern of fold-line  $KPEW$  in the course of point  $W$  tracking the curve  $WW_m W_e$  as shown in Fig. 14. It is clear that no overlapping exists between the spatial region scanned by the fold-line  $KPEW$  and the wall surface, implying that the mobile manipulator can successfully carry out the window operation with the said algorithm.

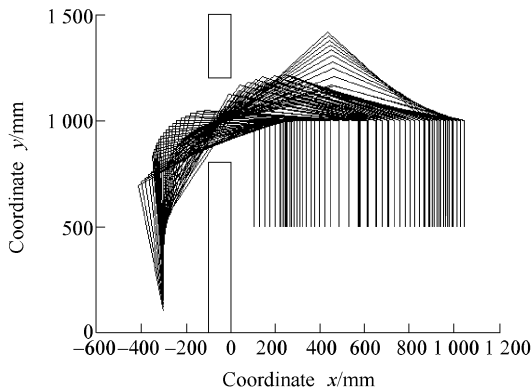


Fig. 14. Diagram of robot's configuration changing pattern in the course of window operation

## 5 Comparative Analysis between the Algorithm and GPM

Since the gradient projection method is an inverse solution optimization in velocity level, it can achieve a satisfying joint velocity only if all the individual coefficients are adjusted appropriately. However, in practical application, the adjustment of coefficients in GPM proves rather challenging, which often make algorithm unstable.

Make use of gradient-projection method depicted in Eq. (1), and then adjust  $\psi$ ,  $k_p$ ,  $\lambda$  according to the magnification factor  $k$  proposed in Eq. (2), however, the ideal joint velocity is unreachable. If set  $\psi = -1$ ,  $k_p = 1$ ,  $\lambda = 40$ , the simulation mentioned in section 3.2.3 is implemented via gradient projection method, where the expected point  $W$  moves along line  $WW'$  to point  $W'$  with uniform speed. As sampling period is 10 ms, the position error curve of end point is shown in Fig. 15, which explains by itself that selection of even smaller sample

period can unavoidably result in a larger position accumulated error in the GPM.

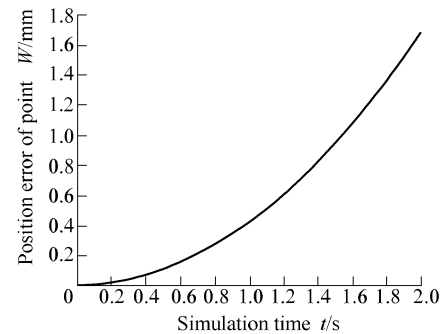


Fig. 15. Distance error change of end track and desired value

In terms of time consumption, we employ Dual CPU with dominant frequency of 2.0 GHz under MATLAB software environment for calculation, divide the line segment  $WW'$  into 200 parts for tracing interpolation, and in simulation the subprogram is supposed to generate every operation and  $\nabla H_v$  in Fig. 1. In contrast to 0.328 s totally consumed in the GPM, to conduct the whole section tracking with the algorithm proposed in this paper, the time consumed turns out to be merely 0.235 s, a slight predominance over the former, which could be explained by the reduced computation load as a result of the spared effort in computing pseudo inverse in the process of tracking. However, for the latter, the computation is done in two steps, meaning the increased number of variable assignment. The improved GPM proposed in Ref. [4] is also tried to conduct simulation, in which the time consumed being 0.623 s when the position tracing precision is set on 0.1 mm. Further investigation is advisably conducted to compare the time consumed for the three methods in performing additional tasks for other redundant manipulators.

## 6 Conclusions

In this paper, we propose a novel motion planning algorithm for redundant manipulator, which is developed by carrying out inverse solution optimization searching directly on the displacement level by means of self-motion manifold. The said algorithm overcomes the disadvantage of terminal displacement error accumulation existing in the GPM. The change of joint velocity tends to be more stable when effectively performing the task of end-effector trajectory tracking and the avoidance of robot link collision simultaneously, which makes it more suitable for the redundant manipulator to accomplish the additional task of collision avoidance on the displacement level. The efficiency of this algorithm has been sufficiently tested and proven by the simulation result.

In terms of computation load, the Jacobian matrix and pseudo-inverse could be disregarded for the algorithm in

question compared to the modified gradient projection; in contrast to other numerical methods, iteration is unnecessary for this algorithm. As a result, when the self-motion manifold of the robot could be approached in analytical method and does not seem over complex, theoretically the computation load of this algorithm will be drastically reduced in comparison with GPM and other numerical methods, which has been testified partly in section 2.2.4. Therefore, the decision is open to the subscribers for this algorithm depending upon the difficulty involved in self-motion manifold solution.

With respect to coefficient adjustment, it is concluded from Eq. (2) and Eq. (3) that the scale factor in this algorithm will only affect redundant joint velocity, which implies that the adjustment work is much easier than the GPM.

## References

- [1] GAO Tongyue, DAI Ju. Research on multi-index amalgamation optimization of redundant robots[J]. *Journal of Huazhong University of Science and Technology*, 2004, 32(10): 71–73. (in Chinese)
- [2] WU Ruimin, LIU Tingrong. New Gradient projection method of kinematically redundant manipulators[J]. *Chinese Journal of Mechanical Engineering*, 1999, 35(1): 76–79. (in Chinese)
- [3] ZU Di, WU Zhenwei, TAN Dalong. Efficient inverse kinematic solution for redundant manipulators[J]. *Chinese Journal of Mechanical Engineering*, 2005, 41(6): 71–75. (in Chinese)
- [4] ZHAO Jianwen. *Research on mechanism and motion planning of P2S5 redundant robot with series/parallel hybrid configuration*[D]. Harbin: Harbin Institute of Technology, 2007. (in Chinese)
- [5] MAYORGA Rend V, CHANDANA Sandeep. A neuro-fuzzy approach for the motion planning of redundant manipulators[C]//*2006 International Joint Conference on Neural Networks*, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, July 16–21, 2006: 2 873–2 878.
- [6] KOKER R I. Reliability-based approach to the inverse kinematics solution of robots using Elman's networks[J]. *Engineering Applications of Artificial Intelligence*, 2005, 18 (3): 685–693.
- [7] OZBAY U, SAHIN H T, ZERGEROGLU E. Robust tracking control of kinematically redundant robot manipulators subject to multiple self-motion criteria[J]. *Robotica*, 2008, 26(6): 711–728.
- [8] YE Ping, SUN Hanxu, JIA Qingxuan. Study on kinematics optimization of redundant manipulators[C]//*IEEE Conference on Robotics, Automation and Mechatronics*, Bangkok, Thailand, Jun 7–9, 2006: 1–6.
- [9] MULLER A. Collision avoiding continuation method for the inverse kinematics of redundant manipulators[C]//*Proceedings – 2004 IEEE International Conference on Robotics and Automation*, New Orleans, USA, Apr 26–May 1, New Orleans: Institute of Electrical and Electronics Engineers Inc., 2004: 1 593–1 598.
- [10] ZHAO Jianwen, DU Zhijiang, SUN Lining. Self-motion manifolds of a 7-DOF redundant robot arm[J]. *Chinese Journal of Mechanical Engineering*, 2007, 43(9): 132–137. (in Chinese)

## Biographical notes

YAO Yufeng, born in 1978, is currently a PhD candidate in State Key Laboratory of Robotics Technique and System, Harbin Institute of Technology, China. He received his bachelor degree from Harbin Institute of Technology, China, in 2002. His research interests include medical equipment and key technology of robot for assisting elderly and handicapped people.

Tel: +86-631-5687811; E-mail: hitwhyf@hitwh.edu.cn

ZHAO Jianwen, born in 1978, is currently a lecturer in Harbin Institute of Technology, China. He received his PhD degree on mechanical electric in Harbin Institute of Technology, China, in 2007.

Tel: +86-631-5687811; E-mail: yipiaowind@163.com

HUANG Bo, born in 1974, is currently an associate professor in Harbin Institute of Technology, China. He received his PhD degree on mechanical electric in Harbin Institute of Technology, China, in 2007.

Tel: +86-631-5672167; E-mail: huangbo74@163.com