2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

An Analytical Algorithm with Minimum Joint Velocity Jump for Redundant Robots in the Presence of Locked-Joint Failures

Zhao Jing, and Li Qian

Abstract—**The joint velocity jump for redundant robots in the presence of locked-joint failures is discussed in this paper. First, the analytical formula of the optimal joint velocity with minimum jump is derived, and its specific expressions for both all joint failure and certain single joint failure are presented. Then, the jump difference between the minimum jump solution and the least-norm velocity solution is mathematically analyzed, and the influence factors on this difference are also discussed. Based on this formula, a new fault tolerant algorithm with the minimum jump is proposed. Finally simulation examples are implemented with a planar 3R robot and a 4R spatial robot, and an experimental study is also done. Study results indicate that the new algorithm proposed in this paper is well suited for real time implementation, and can further reduce the joint velocity jump thereby improving the motion stability of redundant robots in fault tolerant operations. Also, the fewer the possible failed joints are, the more obvious the effect of this new algorithm is.**

I. Introduction

When robots perform a task in harsh and/or remote environments, they are subject to actuator and sensor failures. For these cases, it is quite difficult or impossible for us to repair the failures on the spot without delays. Therefore, the tolerance to failures is essential for the robots to carry out such tasks as space exploration [1], underwater exploration [2] and hazardous material disposal [3]. The so-called fault tolerance means that a robot can still continue the desired tasks in the presence of failures. The failure modes of a robot include locked-joint, free-swinging joint and following-motion joint [4], where the locked-joint is one of the most common modes. The locked-joint failure will be discussed in this article. This failure covers two cases: one is active locking, where a joint can be locked by fail-safe brakes when a robot is capable of detecting its failure in advance; the other is passive locking, where a joint is locked unexpectedly due to mechanical failures.

Since redundant robots have "extra" degree-of-freedom (DOF) which can compensate the motions of failed joints to continue the desired tasks after failures, they are often used in fault tolerant operations [5]. When any joint of a redundant robot with one redundancy fails and is locked, the robot will degrade to a new robot with different structural parameters that is called reduced robot here. Obviously, the performance of the robot will inevitably degrade. The evaluation on the performance of a reduced robot is one of significant problems. The minimum singular value of the reduced Jacobian matrix and the reduced manipulability of the robot are two of common fault tolerant indexes, which can be used to evaluate the dexterity of the reduced robot at the instant of locking failed joint [5]-[7]. The volume of the fault tolerant workspace and the centrality index that describes the positional relation between the fault tolerant workspace and the operational task are the other two fault tolerant indexes, which are often used to quantify the dexterity of the reduced robot during post-failure operations [8]-[10]. Besides the dexterity of reduced robots, the joint velocity jump at the instant of locking failed joint is another critical issue that should have been studied [11]. At this instant the reduced robot will replace the healthy robot to continue the desired task. To exactly follow the desired trajectory of the end-effector, the velocities and torques of surviving joints in the reduced robot will produce a jump, which inevitably reduces the accuracy of the end-effector trajectory. This problem was addressed to some extent. Reference [12] proposed a fault tolerant control algorithm based on the minimum singular value of the reduced Jacobian matrix. This algorithm can avoid the singularity of the reduced robot to ease the joint velocity jump. Nevertheless, research results show that the jump inherently results from the structural difference between the healthy robot and the reduced robot, and so it occurs even if the reduced robot is not in singular configurations [13].

A fault tolerant algorithm with minimum joint velocity jump was proposed [13], [14]. However, its computation is heavy, and not suitable for real time control. The remainder of this paper is organized as follows. Section II defines the joint velocity jump. In section III, the analytical formula of the optimal joint velocity with minimum jump is derived, and its specific expressions for both all joint failure and certain single joint failure are presented. Besides some problems related to this formula including a comparison with the least-norm velocity solution, singularity and a new fault tolerant planning algorithm are also discussed in this section. Simulation examples of a planar 3R robot and a spatial 4R robot are presented in Section IV. An experimental study is done in section V. Finally, the conclusions are given in Section VI.

II. Definition of The Joint Velocity Jump

Assuming that a robot has n DOF and m absolute parameters of the end-effector. When one of robot's joints fails and is locked, the robot will change into a reduced robot

The support of this work by National Natural Science Foundation of China (50775002) and Funding Project for Academic Human Resources Development in Institutions in Higher Learning under the Jurisdiction of Beijing Municipality is greatly appreciated.

Zhao Jing and Li Qian are with the College of Mechanical Engineering and Applied Electronics Technology, Beijing University of Technology, Beijing 100022, China (e-mail: zhaojing@bjut.edu.cn; leeqian020112@yahoo.com).

and the velocity of surviving joints will also be redistributed due to the failed and locked joint. In this way, the difference of joint velocity between the healthy robot and the reduced robot may occur. The difference is defined as joint velocity jump (JVJ) and formulated as follows [13].

$$ {}^{i}\lambda_j = \left| {}^{i}\dot{\theta}_j - \dot{\theta}_j \right| \tag{1} $$

where $\dot{\theta}_j$ is the joint velocity of the healthy robot, and ${}^{i}\dot{\theta}_j$ is the joint velocity of the reduced robot. Subscript "j" ranges from 1 to n, which denotes all joints; superscript "i" denotes the failed joints. When $j = i$, ${}^{i}\lambda_j = \left|\dot{\theta}_j\right|$ since the failed joint is locked, i.e. ${}^{i}\dot{\theta}_j = 0$. It is evident that the JVJ depends on not only the joint velocity of the healthy robot but also that of the reduced robot. The smaller ${}^{i}\lambda_j$ is, the smaller the JVJ is, and namely, the failed joint has smaller influence on the kinematical properties of a robot. This means that the operational accuracy of the robot at the instant of failures is higher. Researches show that the excessive JVJ will result in vibration and jerk to a robot, which does reduce its operational accuracy at the instant of failures [13],[14].

In (1), ${}^{i}\dot{\theta}_j$ is decided by the reduced Jacobian matrix, which is given by the original Jacobian matrix $\boldsymbol{J}$ with its i-th column removed. Thus, the reduced Jacobian matrix can be expressed: $\boldsymbol{J}_i = \left[\boldsymbol{j}_1, \cdots, \boldsymbol{j}_{i-1}, \boldsymbol{j}_{i+1}, \cdots, \boldsymbol{j}_n\right]$. In the case of redundant robots with one redundancy, the ${}^{i}\dot{\theta}_j$ can be calculated by

$$ {}^{i}\dot{\boldsymbol{\theta}} = \boldsymbol{J}_i^{-1}\dot{\boldsymbol{X}} \tag{2} $$

where $\dot{\boldsymbol{X}} \in \boldsymbol{R}^{m\times 1}$ is the velocity vector of the end-effector, and $\boldsymbol{J}_i^{-1} \in \boldsymbol{R}^{m\times(n-1)}$ is the inverse of the reduced Jacobian.

III. Analytical Formula of The Joint Velocity and Related Problems

A. Derivation of the Formula

Equation (1) indicates that the JVJ depends on the difference of joint velocity between the healthy robot and the reduced robot. For the redundant robot with one redundancy, when any joint of the robot fails and is locked, the joint velocities of the reduced robot are uniquely determined by the desired trajectory of the end-effector. For this reason, one can optimize the joint velocities of the healthy robot through the joint self-motion to make it get close to that of the reduced robots as much as possible. This optimal motion planning can be stated as: on condition that the end-effector's motion is satisfied, optimize the joint velocities of the redundant robot to minimize the square sum of the JVJ. Assuming that any redundant joint of a robot is possible to fail, the optimal motion planning can be formulated as

$$ \text{Min Z} = \sum_{i=1}^{n} \frac{1}{2}({}^{i}\dot{\boldsymbol{\theta}}' - \dot{\boldsymbol{\theta}})^{\mathrm{T}}({}^{i}\dot{\boldsymbol{\theta}}' - \dot{\boldsymbol{\theta}}) \tag{3} $$

Subject to

$$ \dot{\boldsymbol{X}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{4} $$

By solving this optimization problem, a general analytical formula of the joint velocity that minimizes the JVJ of the redundant robot can be obtained as

$$ \dot{\theta} = \boldsymbol{J}^{+}\dot{\boldsymbol{X}} + \frac{1}{n}(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})(\sum_{i=1}^{n} {}^{i}\dot{\theta}') \tag{5} $$

where $\boldsymbol{J}^{+} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1} \in \boldsymbol{R}^{n\times m}$ is the Moore-Penrose generalized inverse of $\boldsymbol{J}$, $\boldsymbol{I} \in \boldsymbol{R}^{n\times n}$ is identity matrix, and "n" is the numbers of possible failed joints.

${}^{i}\dot{\boldsymbol{\theta}}' = [{}^{1}\dot{\theta}_1, \cdots, {}^{i}\dot{\theta}_{i-1}, 0, {}^{i}\dot{\theta}_{i+1}, \cdots, {}^{i}\dot{\theta}_n]^{\mathrm{T}} \in \boldsymbol{R}^{n\times 1}$ is the extended joint velocity vector of the reduced robot, whose non-zero components can be calculated by (2).

Equation (5) shows that the optimal joint velocity with minimum JVJ can be written in the standard pattern of the gradient projection algorithm [15]. The specific expression of the sum of the extended joint velocity vector depends on the numbers of possible failed joints.

For a robot with one degree of redundancy, if all joints are possible to fail, that is, on the condition that all single joint failure is considered, the sum of the extended joint velocity vector of the reduced robot can be calculated by

$$ \sum_{i=1}^{n} {}^{i}\dot{\boldsymbol{\theta}}' = \begin{bmatrix} 0 \\ {}^{1}\dot{\theta}_2 \\ \vdots \\ {}^{1}\dot{\theta}_n \end{bmatrix} + \begin{bmatrix} {}^{2}\dot{\theta}_1 \\ 0 \\ \vdots \\ {}^{2}\dot{\theta}_n \end{bmatrix} + \ldots + \begin{bmatrix} {}^{n}\dot{\theta}_1 \\ {}^{n}\dot{\theta}_2 \\ \vdots \\ 0 \end{bmatrix} \tag{6} $$

If only joint k is possible to fail and is locked, (6) then becomes

$$ \sum_{i=1}^{n} {}^{i}\dot{\boldsymbol{\theta}}' = [{}^{k}\dot{\theta}_1, \cdots, {}^{k}\dot{\theta}_{i-1}, 0, {}^{k}\dot{\theta}_{i+1}, \cdots, {}^{k}\dot{\theta}_n]^{\mathrm{T}} \tag{7} $$

In this case, assuming that joint "n" is locked, an easy derivation will arrive at

$$ \dot{\boldsymbol{\theta}} = \boldsymbol{J}^{+}\dot{\boldsymbol{X}} + \begin{bmatrix} \boldsymbol{J}_n^{-1}\dot{\boldsymbol{X}} \\ 0 \end{bmatrix} - \boldsymbol{J}^{+}\left[\boldsymbol{J}_n \boldsymbol{j}_n\right] \begin{bmatrix} \boldsymbol{J}_n^{-1}\dot{\boldsymbol{X}} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_n^{-1}\dot{\boldsymbol{X}} \\ 0 \end{bmatrix} \tag{8} $$

This result indicates that, on the condition that only one of joints is possible to fail and is locked, to minimize the JVJ, the joint should be locked in advance. In this way, the joint velocities of the original robot are equal to those of the reduced robot, and so the JVJ keeps zero throughout. Obviously, this is only an extreme case. For most cases, all the joints, at least some of joints, are possible to fail. Hence, the joint velocity is usually not equal to zero.

B. Comparison with the Least-norm Velocity Solution

The least-norm velocity solution can operate a robot with the minimum joint velocity in square sum sense. Generally, the larger the velocity of the failed joint is, the larger the JVJ of the robot is. This implies that except the optimal joint velocity solution with minimum JVJ based on (5), the least-norm velocity solution is better than other solutions with respect to the JVJ index. It is therefore a natural question: What is the difference of the JVJ between the least-norm solution and the minimum JVJ solution? And, what are the key influence factors on the difference? They will be answered in this section.

When a redundant robot operates in terms of the least-norm solution and the minimum JVJ solution respectively,

assuming that all single joint failure is considered, their square sum of the JVJ can be respectively expressed as

$$\lambda_{LN} = \sum_{i=1}^{n} \left\| {}^{i}\dot{\boldsymbol{\theta}}' - \dot{\boldsymbol{\theta}}_{LN} \right\|^2 \tag{9}$$

$$\lambda_{OPT} = \sum_{i=1}^{n} \left\| {}^{i}\dot{\boldsymbol{\theta}}' - \dot{\boldsymbol{\theta}}_{OPT} \right\|^2 \tag{10}$$

where $\|\bullet\|$ is 2 norm of joint velocity vector, $\dot{\boldsymbol{\theta}}_{LN}$ is least-norm joint velocity solution, and $\dot{\boldsymbol{\theta}}_{OPT}$ is optimal solution with minimum JVJ.

From (6), the minimum JVJ solution can be written

$$\dot{\boldsymbol{\theta}}_{OPT} = \dot{\boldsymbol{\theta}}_{LN} + \dot{\boldsymbol{\theta}}_{H} \tag{11}$$

where $\dot{\boldsymbol{\theta}}_{H}$ is a homogenous solution belonging to the null space, and orthogonal to $\dot{\boldsymbol{\theta}}_{LN}$.

Expanding (11) and collecting terms lead to

$$\lambda_{LN} = \mathrm{n}\sum_{j=1}^{n} \dot{\theta}_{LN_j}^2 + \sum_{j=1}^{n}\sum_{i=1}^{n} {}^{i}\dot{\theta}_j^2 - 2\sum_{j=1}^{n}(\dot{\theta}_{LN_j}\sum_{i=1}^{n} {}^{i}\dot{\theta}_j) \tag{12}$$

In similar manner, substituting (11) into (10), and noting the orthogonality of $\dot{\boldsymbol{\theta}}_{H}$ and $\dot{\boldsymbol{\theta}}_{LN}$, we obtain

$$\begin{aligned}\lambda_{OPT} = \mathrm{n}\,(\sum_{j=1}^{n} \dot{\theta}_{LN_j}^2 - \sum_{j=1}^{n} \dot{\theta}_{H_j}^2) + \sum_{j=1}^{n}\sum_{i=1}^{n} {}^{i}\dot{\theta}_j^2 \\ - 2\sum_{j=1}^{n}(\dot{\theta}_{LN_j}\sum_{i=1}^{n} {}^{i}\dot{\theta}_j)\end{aligned} \tag{13}$$

Thus, the difference δ in the square sum of the JVJ between the two solutions can be expressed as

$$\delta = \lambda_{LN} - \lambda_{OPT} = \mathrm{n}\sum_{j=1}^{n} \dot{\theta}_{H_j}^2 \tag{14}$$

Evidently, (14) is always greater than zero. This shows that in the case of the jump index the minimum JVJ solution does outperform the least-norm solution; and the difference depends on the homogenous solution of the joint velocity for a redundant robot and its numbers of possible failed joints. Generally, the fewer the possible failed joints are, the greater the difference is (noting that $\dot{\boldsymbol{\theta}}_{H}$ consists of 1/n).

It is worth explaining that the least-norm solution and the minimum JVJ solution all belong to a local solution and the comparison of joint velocities between the two solutions is also based on a specific joint configuration. Therefore, above conclusion is exactly true only when the two solutions have the same joint configuration. Nevertheless, the end-effector of a robot is often required to follow a continuous trajectory. At this time, if the two solutions are respectively used to generate a family of joint trajectories, even at the same instant, because the robot's joint configuration is different, (14) is not satisfied, i.e. δ is not always greater than zero.

C. Singularity of A Reduced Robot

When one of robot's joints fails and is locked, the robot will change to a reduced robot with fewer DOF, and its D-H (Denavit-Hartenberg) parameters will also vary with the position where the failed joint is locked. Since robot failures may occur at random anywhere over its entire range of motion, all the reduced robots at any instant of possible failures must be non-singular and more dexterous. The minimum singular value and the reduced manipulability based on the reduced Jacobian matrix are commonly used to quantify the dexterity of a redundant robot at the instant of failures [5], [6]. These indexes have a dimension relative to length unit. To obtain a dimensionless index, the condition number of a robot can be introduced [16]. Therefore, we now define the reduced condition number of the Jacobian (RCN) as a new fault tolerant index, and sign it with k_i

$$k_i = \sigma_{1i} / \sigma_{ri} \tag{15}$$

where, σ_{1i} and σ_{ri} are the maximum and minimum singular values of the reduced Jacobian matrix $\boldsymbol{J}_i$ respectively. Obviously, $1 \le k_i < \infty$. The smaller the k_i is, the stronger the dexterity of the reduced robot with locked joint "i" is, which means the robot has stronger fault tolerant ability at the instant of failures. When k_i =1, the reduced robot is isotropic and the most dexterous. If the fault tolerance of more than one joint is considered, the square sum of the dexterity for every joint can be taken, which is consistent with the JVJ index discussed above. Using the gradient of the RCN to replace the sum of the extended joint velocity vector in (5), we can obtain a fault tolerant algorithm with minimum dexterity.

In most cases, although (5) can minimize the JVJ, it can not guarantee the reduced robot to avoid singular configurations. For a given task, the joint trajectories of a robot are related to the initial postures of its end-effector. Different initial postures correspond to different solution domains of the joint motion. Therefore, properly choosing the initial postures can avoid the singular configurations of a robot while minimizing the JVJ.

D. A Fault Tolerant Planning Algorithm with the JVJ

The calculation procedures of this algorithm are as follows:

(i) Arbitrarily choose the initial position of the end-effector, and let the end-effector's velocity be zero. Determine the optimal initial configuration with minimum RCN through the joint self-motion in the null space [17].

(ii) Determine the joint trajectories under this initial condition, and calculate the RCN at any instant.

(iii) Compare the calculated RCN with a specified threshold value. If the RCN is greater than the threshold value, the calculation ends; otherwise, return to step 1 until the requirement is satisfied.

It is indicated that when (5) is adopted to minimize the JVJ in the fault tolerant motion planning, the singular configurations can be avoided by adjusting the end-effector's initial position. Since our main interests in this paper are the JVJ problem of a robot but not its singularity problem, and the approach to choose the initial position to avoid the singularity of the robot was discussed in [13], this new algorithm does not include how to determine an optimal initial position for the robot.

IV. Simulation Examples

A. Planar 3R Robot

Fig. 1 is a planar 3R robot. For positional tasks it is a redundant robot with one degree of redundancy. The three

algorithms including the JVJ algorithm (JVJA), the least norm velocity algorithm (LNVA) and the maximum dexterity algorithm (MDA) will be respectively implemented with this robot for the following two cases: all joint failures and two joint failures. And the comparisons of these algorithms on the JVJ and the RCN indexes will also be made. The simulation conditions are as follows. The three links of the robot are identical and each is 0.5 m long. Assume the velocity of its end-effector $\dot{\boldsymbol{X}} = [0.2\pi\sin(2\pi t), 0.2\pi\cos(2\pi t)]^{\mathrm{T}}$ *m/s* , simulation time t=1.0 s and calculation time dt=0.001 s.

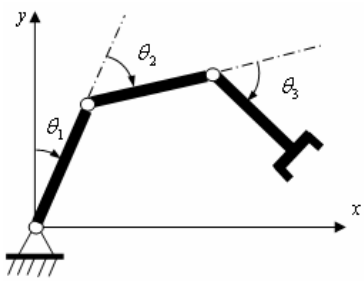
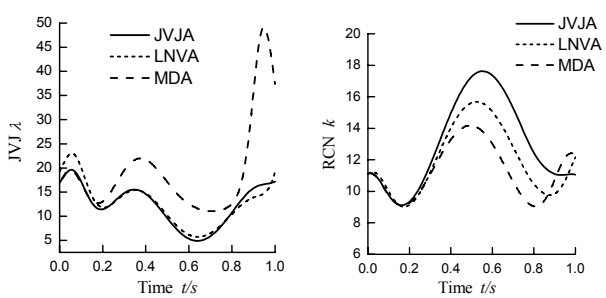


Fig. 1. A Planar 3R Robot.

1) All Joint Fault Tolerance: Fig. 2(a) and (b) are the simulation results obtained respectively by the JVJA, the LNVA and the MDA, where the initial position of the end-effector $\boldsymbol{X} = [0.35,0]^{\mathrm{T}}$ m , and the two indexes, λ and k are in their square sum sense. It is seen that on the joint velocity jump the JVJA is the best while the MDA is the poorest; on the dexterity, it is quite reverse. The two indexes of the LNVA are in the middle of the three algorithms.

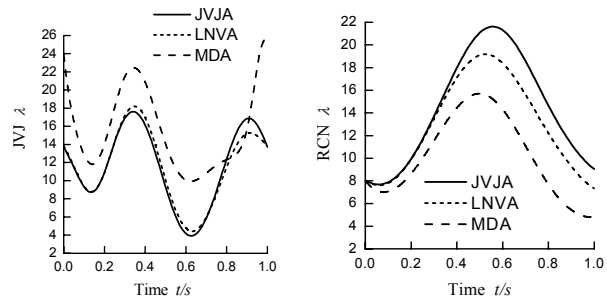
It is noted that the solutions may cross in quality at a certain point: for instance, the MDA solution in Fig. 2(b) grows over the ones with JVJA and LNVA at t=0.85. This is due to the local characteristics of these algorithms as motioned above.



(a) (b)

Fig. 2. Fault tolerant indexes of the three algorithms for all joint fault tolerance when initial position of the end-effector $\mathbf{X} = [0.35,0]^{\mathrm{T}}$ m . (a) Joint Velocity Jump, (b) Reduced Condition Number.

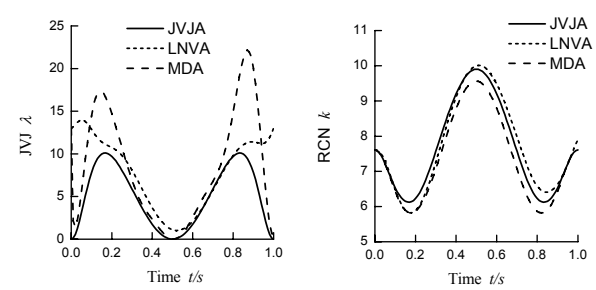
Next, we change the initial position of the end-effector and let $\boldsymbol{X} = [0.4,0]^{\mathrm{T}}$ m . Similar simulation results are presented in Fig. 3(a) and (b). It is shown that the initial positions of the end-effector affect the absolute values of the JVJ and the RCN of these algorithms, but almost do not affect the relative values of the two indexes. Generally, the two indexes of these algorithms conflict with each other, i.e., adjusting the initial position of the end-effector can improve one index, meanwhile it will worse the other. In addition, for the all joint fault tolerance of a robot, no matter where the initial position is, the LNVA is always close to the JVJA on the jump index, and the closing extent is related to the D-H parameters of the robot.



(a) (b)

Fig. 3. Fault tolerant indexes of the three algorithms for all joint fault tolerance when initial position of the end-effector $\mathbf{X} = [0.4,0]^{\mathrm{T}}$ m . (a) Joint Velocity Jump, (b) Reduced Condition Number.

2) Two Joint Fault Tolerance: If only joints 2 and 3 are possible to fail, and the initial position of the end-effector is taken as $\boldsymbol{X} = [0.35,0]^{\mathrm{T}}$ m , we can obtain the simulation results shown in Fig. 4(a) and (b). These results indicate for the jump index the optimal effect of the JVJA in this case is more obvious than in all joint fault tolerance; but for the RCN, there is no big difference among the three algorithms. It is still true for other initial positions. This means that the JVJA is more effective if only some of joints are possible to fail. Besides, the influence of the initial position of a robot's end-effector on the two indexes is similar to the case of the all joint fault tolerance.



(a) (b)

Fig. 4. Fault tolerant indexes of the three algorithms for two joint fault tolerance. (a) Joint Velocity Jump, (b) Reduced Condition Number.

B. Spatial 4R Robot

A spatial 4R robot and its D-H coordinate system are given in Fig. 5. It is also a redundant robot with one degree of redundancy relative to a positional task. Assume the D-H parameters d_1, d_2, d_3, d_4, and d_t to be 0.5m, 0m, 0.4m, 0.1m, and 0.3m respectively; the velocity of the end-effector $\dot{\mathbf{X}} = [0, 0.15\pi\sin(2\pi\ \mathrm{t}), 0.15\pi\cos(2\pi\ \mathrm{t})]^{\mathrm{T}}$ m/s , the other simulation conditions are the same as that of the planar 3R robot. And only all joint fault tolerance is discussed here. It is noted that the so-called all joints actually only include joints 1, 2, and 3 since the joint 4 of this robot is a non-redundant joint, whose motion is uniquely decided by the end-effector's motion.

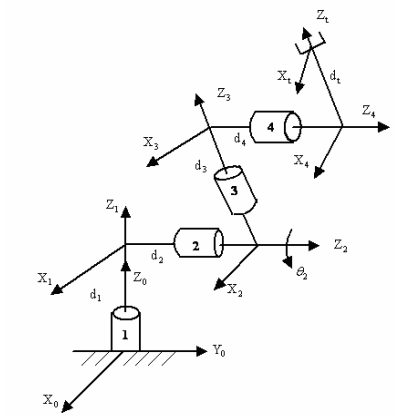
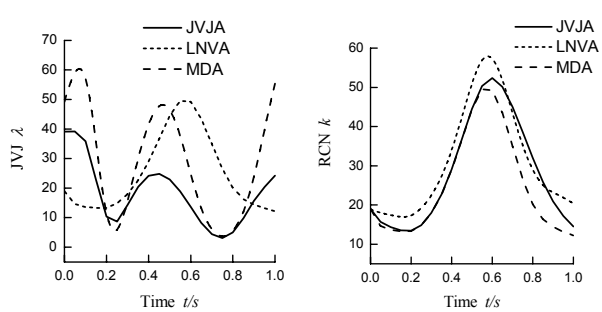


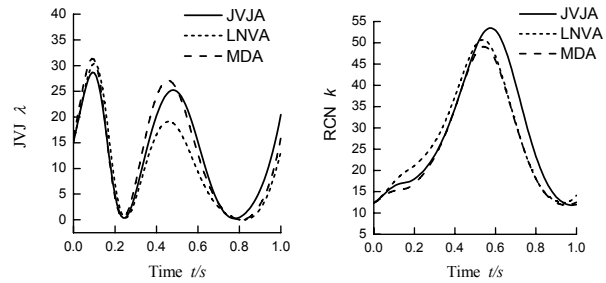
Fig. 5. A Spatial 4R Robot.

To compare with the planar 3R robot, two different initial positions of the end-effector, $\boldsymbol{X} = [0.09, 0.21, 0.27]^{\mathrm{T}}$ m and $\boldsymbol{X} = [-0.21, 0.2, 0.35]^{\mathrm{T}}$ m are taken to do simulations, whose results are presented in Fig.6-Fig.7. At the first initial position, the jump index of the JVJA is much smaller than that of the other two algorithms. Meanwhile its reduced condition number is quite close to that of MDA. However at the second initial position, the three algorithms are all quite close on the two indexes. This phenomenon mainly results from the local characteristics of these algorithms. Vast simulation results show that the fault tolerant planning effects of spatial robots are more sensitive to the initial positions than that of planar robots. It is because spatial robots possess stronger non-linear properties on kinematics. In the case of spatial robots, the initial positions of the end-effector affect not only the absolute values of the JVJ and the RCN of these algorithms, but also the relative values of the two indexes, which is essentially different from the planar robot case. This implies that when the JVJA is applied in the fault tolerant planning of spatial robots, on one hand, the initial positions have to be carefully chosen, otherwise, the desired optimal effects can't be obtained; on the other hand, this new algorithm can provide much better joint trajectories by optimizing the initial positions. By optimizing the initial positions the global characteristics of the algorithms can be improved [13].



(a) (b)

Fig. 6. Fault tolerant indexes of the three algorithms for all joint fault tolerance when initial position of the end-effector $\mathbf{X} = [0.09, 0.21, 0.27]^{\mathrm{T}}$ m . (a) Joint Velocity Jump, (b) Reduced Condition Number.



(a) (b)

Fig. 7. Fault tolerant indexes of the three algorithms for all joint fault tolerance when initial position of the end-effector $\mathbf{X} = [-0.21, 0.2, 0.35]^{\mathrm{T}}$ m . (a) Joint Velocity Jump, (b) Reduced Condition Number.

V. An Experimental Study

In this section, an experimental study is done with a Power-cube planar 3R robot shown in Fig. 9 to demonstrate that the influence of joint velocity jump on the operational accuracy of robot's end-effector at the moment of locking failed joint. The test bed shown in Fig. 8 consists of a Power-Cube planar 3R robot and Optotrak 3D motion measurement system, where the three links of the robot are 0.29m, 0.32m and 0.23m long, and the end-effector's motion is prescribed by $\dot{\mathbf{X}} = [-(\pi/30)\sin(\pi t/3), (\pi/30)\cos(\pi t/3)]^{\mathrm{T}}$ m/s. It is assumed that only joint 3 is possible to fail and to be locked at the moment of maximum and minimum jump respectively.

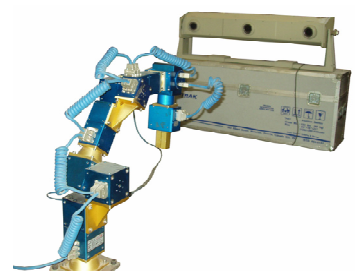


Fig. 8. Power-cube Planar 3R Robot

For convenience and not losing generality, the experiment is implemented with the least norm velocity algorithm. The minimum and maximum jumps occur when t=4.5 s and t=3.1s respectively. Fig.9 and Fig.10 are the paths of the end-effector and their errors when joint 3 is locked at the moments of minimum and maximum jumps respectively. Fig.10 shows that in the two cases the robot all can still finish the desired circle path even in the presence of locking-joint failures. However, the jump of path error at the maximum jump moment is far greater than that at the minimum jump moment as shown in Fig.10, where the path error is measured by the difference between its least square values and actual values. These results show that the motion stability of redundant robots in fault tolerant operations can be effectively improved by reducing the joint velocity jump at the moment of locking failed joints. It is therefore concluded

that the new algorithm with a lower joint velocity jump can increase the accuracy of the end-effector in fault tolerant operations.

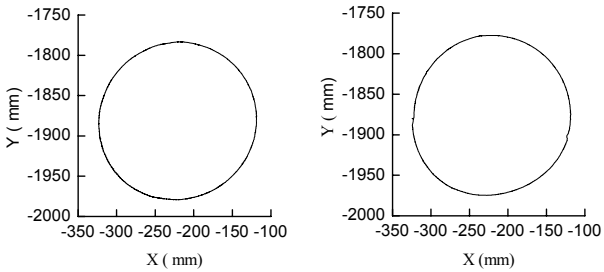


Fig. 9. The Path of the robot's end-effector . (a) At Minimum Jump, (b) At Maximum Jump.

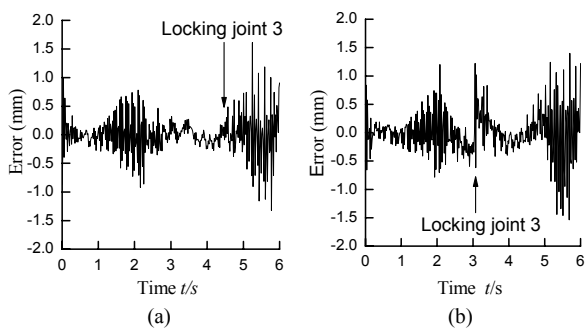


Fig.10. The Path errors of the robot's end-effector. (a) At Minimum Jump, (b) At Maximum Jump.

VI. CONCLUSIONS

The research results indicate in comparison with the LNVA and the MDA, the new algorithm proposed can reduce the joint velocity jump at the instant of locking failed joint, and, the fewer the possible failed joints are, the more obvious the effect of this new algorithm is. In the case of planar robots, the initial positions of the end-effector affect the absolute values of the JVJ and the RCN of these algorithms, but almost do not affect the relative values of them on the two indexes. However, in the case of spatial robots, the initial positions of the end-effector affect not only the absolute values of the JVJ and the RCN of these algorithms, but also the relative values of the two indexes. That is to say, the fault tolerant planning effects of spatial robots are more sensitive to the initial positions than that of planar robots, which is an essential difference between spatial robots and planar robots.

In addition, a study indicates that, in the case of a spatial robot the joint layout and the D-H parameters affect the optimal effect of the new algorithm more seriously. It is not difficult to imagine that, for a given operational task, there at least exists one robot with a specific joint layout and corresponding D-H parameters that can minimize the joint velocity jump while satisfying the RCN requirement. How to determine the joint layout and the D-H parameters, that is, the dimensional synthesis of a robot based on the two fault tolerant indexes is our next step in the future researches.

REFERENCES

[1] G. Visentin and F. Didot, "Testing space robotics on the Japanese ETS-VII satellite," ESA Bulletin-European Space Agency, pp. 61-65, September, 1999.

[2] P. S. Babcock and J.J. Zinchuk, "Fault-tolerant design optimization: Application to an autonomous underwater vehicle navigation systems," In Proceeding Symp. Autonom. Underwater Vehicle Technol., Washington, D.C., pp. 34-43, 1990.

[3] M.L. Leuschen, I.D.Walker and J. R. Cavallaro, "Investigation of reliability of hydraulic robots for hazardous environment using analytic redundancy," In Proceeding Annual Rel. Maintain. Symp., Washington, D.C., pp.122-128, 1999.

[4] J.D. English and A.A. Maciejewski, "Failure tolerance through active braking: A kinematic approach," Int. Journal of Robotics Research, vol.20, no.4, pp.287-299, 2001.

[5] A.A. Maciejewski, "Fault Tolerant properties of kinematically redundant manipulator," In Proceeding IEEE International conference on Robotics and Automation, pp. 638-642, 1990.

[6] R.G. Roberts and A.A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," IEEE Transactions on Robotics and Automation, vol.12, no.4, pp. 543-552, 1996.

[7] C.Caldwell and R.G.Roberts, "Fault–tolerant kinematically redundant robots," In Proceeding Thirty-Fourth Southeastern Symposium on system Theory, pp. 281-285, 2002.

[8] C.L.Lewis and A.A.Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," IEEE Transactions on Robotics and Automation , vol.13, no.4, pp. 622-629, 1997.

[9] Zhao Jing and Jing Hongmei, "Fault tolerant motion planning for two coordinating manipulators," Chinese Journal of Mechanical Engineering, vol.40, no.12, pp.172-176, 2004.

[10] M.Goel, A.A. Maciejewsk and V. Balakrishnan, "Analyzing unidentified locked-joint failures in kinematically redundant manipulators," Journal of Robotic systems, vol.22, no.1, pp. 15-29, 2005.

[11] T.Renato, H.T..Marco and B. Marcel, "Fault tolerance in cooperative manipulators," In Proceeding IEEE International conference on Robotics and Automation, Washington, DC, pp. 470-475, May, 2002.

[12] K.N.Groom, A.A. Maciejewski and V. Balakrishnan, "Real-time failure tolerant control of kinematically redundant manipulators," IEEE Transactions on Robotics and Automation, vol.15, no.6, pp.1109-1116, 1999.

[13] Zhao Jing, Yao Xuebin and Zhang Lei, "The Optimization of initial posture with avoidance of the sudden change in joint velocity for fault tolerant operations of two coordinating redundant manipulators," Mechanism and Machine Theory, vol.40, no.6, pp. 659-668, 2005.

[14] Zhao Jing, Zhang Kailiang and Yao Xuebin, " Study on fault tolerant workspace and fault tolerant planning algorithm based on optimal initial position for two spatial coordinating manipulators," Mechanism and Machine Theory, vol.41, no.5, pp. 584-595, 2006.

[15] A.Liegeois, "Automation supervisory control of the configuration and behavior of multi-body mechanisms," IEEE Trans on Syst Man Cybern, vol.27, no.12, pp.868-871, 1977.

[16] J.K. Salibury and J.Craig, "Articulated hands: kinematic and force control," Int. Journal of Robotics Research, vol.1, no.1 1, pp. 4-17, 1982.

[17] P. H. Chang, "A closed-form solution for control of manipulators with kinematic redundancy," In Proceedings IEEE International conference on Robotics and Automation, pp. 722-728, 1985.