Research Article

Trajectory Optimization for Velocity Jumps Reduction considering the Unexpectedness Characteristics of Space Manipulator Joint-Locked Failure

Qingxuan Jia, Tong Li, Gang Chen, Hanxu Sun, and Jian Zhang

School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Tong Li; bupt.litong@gmail.com

Aiming at reducing joint velocity jumps caused by an unexpected joint-locked failure during space manipulator on-orbit operations without shutting down manipulator, trajectory optimization strategy considering the unexpectedness characteristics of joint-locked failure is proposed in the paper, which can achieve velocity jumps reduction in both operation space and joint space simultaneously. In the strategy, velocity in operation space concerning task completion directly is treated as equality constraints, and velocity in joint space concerning motion performance is treated as objective function. Global compensation vector which consists of coefficient, gradient of manipulability, and orthogonal matrix of null space is constructed to minimize the objective function. For each particular failure time, unique optimal coefficient can be obtained when the objective function is minimal. As a basis, a method for optimal coefficient function fitting is proposed based on a priori failure information (possible failure time and the corresponding optimal coefficient) to guarantee the unexpectedness characteristics of joint-locked failure. Simulations are implemented to validate the efficiency of trajectory optimization strategy in reducing velocity jumps in both joint space and operation space. And the feasibility of coefficient function is also verified in reducing velocity jump no matter when joint-locked failure occurs.

1. Introduction

Space manipulator [1, 2] is a kind of complex electromechanical system, which plays an important role in space station maintenance, rendezvous and docking, and space exploration. During its long-time service on orbit, affected by radiation, energetic particle, and high-low temperature alternating of space environment, many kinds of malfunctions are inevitable, such as link deformation, performance degradation, and joint failure [3, 4]. Joint as a precision unit, integrating with machinery, communications, and automatic control technology, is the core component of space manipulator to achieve safe and smooth operation. Joint failure may bring about performance degradation of space manipulator and lead to bad effect on task completion.

For general manipulator on the ground, when joint fails during manipulator motion, the manipulator can be shut down to maintain or replace the failure joint. However, if

shutting down space manipulator once joint failure occurs, the inertia force acting on the manipulator will cause serious impact force on the end-effector and joints, as well as cause instability of the base. When joint fails especially during load operations such as space cabin docking [5] and astronauts assistant [6], the bigger mass and inertia will cause huger impact, which may exceed the rated torque and cause serious hurt on the manipulator parts and operating objects. On the other hand, when trying to execute task continuously after joint failure, control strategy should be adjusted in real time. As a result, velocity jumps will be caused in joint space and operation space, which will also generate impact on manipulator structure. Thus, it is of great significance to smooth motion parameters when joint failure occurs during on-orbit operations no matter whether space manipulator will be shut down or execute task continuously.

Joint failure can be divided into joint-locked [7] and free swing [8] according to whether joint can be operated

or not after failure occurs. In free swing state, failure joint moves freely, making the manipulator out of control. In joint-locked state, failure joint is locked at the angle of failure time. Although the degree of freedom (DOF) of space manipulator is reduced, the healthy joints can move normally, and the task can be executed continuously when joint-locked failure occurs. Thus in order to avoid the impact force caused by shutting down space manipulator after joint-locked failure, the paper is devoted into trajectory optimization research to achieve task completion and motion parameters smoothness (velocity jumps reduction especially) without shutting down manipulator.

In order to achieve task completion after manipulator failure, scholars concentrate on the following researches. First, in order to obtain the failure information such as failure time and failure joint, fault detection, fault isolation, and fault identification [9–12] are researched. Second, abilities of manipulator are analyzed in failure condition such as fault-tolerant workspace [13, 14] and dexterity [15, 16]. Based on these researches above, pseudoinverse method [17], self-motion manifold method [18], quadratic programming algorithm [19], and configuration optimization [20, 21] are applied to achieve task completion after failure occurs. Meanwhile, constraints are analyzed during fault tolerance to guarantee the practical application. Ralph and Pai [22] analyzed the minimal constraints necessary in fault tolerance, Jamisola Jr. et al. [23] took environment obstacle constraints into account, and Xie et al. [24] introduced the singularity constraints into the fault-tolerant method.

In the mentioned fault-tolerant researches, some of them shut down manipulator after failures; then fault-tolerant method is introduced to achieve task completion. However, it is not suitable for space manipulator fault-tolerance. The others execute task via fault-tolerant method in real time. Unfortunately, when guaranteeing velocity and position in operation space for task completion, velocity jumps are brought about into joint space. Scholars try to reduce the velocity jumps; Abdi et al. [25] concentrated on the minimum velocity jump in operation space and proposed an optimization strategy for joint velocity redistribution. Meanwhile, the velocity jump in joint space is reduced with the application of least square method and matrix perturbation method [26]. Jing and Cheng [27] achieved velocity jump reduction in joint space by optimizing the initial configuration of a dual-arm manipulator. In the authors' former work [28], velocity jumps in joint space were reduced by constructing compensation vectors based on the gradient of manipulability. Further, the coefficient form of compensation vector was discussed to achieve better reduction performance.

However, simultaneous reduction for velocity jumps in operation space and joint space is not achieved. What is more, velocity jumps reduction in the operation space may even increase the velocity jumps in joint space. Since velocity jumps in operation space are concerned with task completion and velocity jumps in joint space are concerned with joint structure safety, it is of equal importance to achieve velocity jumps reduction in the two spaces to guarantee task completion and motion safety. According to the self-motion characteristics of space manipulator, a set of velocities in operation

space corresponds to multiple sets of joint velocities. So when the velocity in operation space is fixed, joint velocity can be adjusted to find an optimal value minimizing the jumps. On the basis, the trajectory optimization strategy can be formed by treating velocity in operation space as constraints and treating velocity in joint space as optimization objective.

Besides, the mentioned fault-tolerant methods depend on failure information; only after the failure condition is confirmed can they begin to work. However, in practical application, joint-locked failure occurs unexpectedly. It needs time to obtain failure information with fault detection method, so time delay is inventible between failure discovery and fault-tolerant method application, which makes fault-tolerant performance worse than the ideal condition. Although controllers are designed based on neural networks to achieve real-time fault tolerance [29–31], the adaptation of controllers for rapid response and stable operation is not enough for on-orbit application. Therefore, in order to avoid time delay problem of fault detection method and to guarantee the unexpectedness characteristics of joint-locked failure, global compensation for joint velocity during the entire path planning is proposed in the paper, based on which the trajectory optimization is not relevant to the failure information and can handle an unexpected joint-locked failure.

In conclusion, the trajectory optimization strategy proposed in the paper aims at dealing with three problems. First, the task completion on-orbit is guaranteed without shutting down space manipulator, and the parameters of space manipulator particularly the velocity jumps should be smoothed after joint-locked failure occurs. Second, the reduction for velocity jumps in both operation space and joint space should be achieved simultaneously. It can be achieved by considering velocity in operation space as equality constraints and considering velocity in joint space as optimization objective in the strategy. Third, the avoidance of time delay problem and the unexpectedness characteristics of joint-locked failure should be handled. For this purpose, a global compensation vector is constructed for velocity compensation over the entire path planning task, which makes the optimization strategy irrelevant with the real-time failure information acquisition. In addition, with global compensation for joint velocity, a unique coefficient which makes objective function minimal can be obtained related to a particular failure time. Taking it as a priori failure information, the law of optimal coefficient related to arbitrary failure time is concluded. A coefficient function is obtained to deal with velocity jump reduction caused by unexpected joint-locked failure.

The presented paper is organized as follows: in Section 2, velocity jumps in operation space and joint space caused by joint-locked failure are analyzed. The mathematical expression of trajectory optimization strategy is derived, during which velocity jumps in operation space are expressed as equality constraint and velocity jumps in joint space are expressed as objective function. In Section 3, global compensation vector for joint velocity is constructed, and the solving method is proposed to obtain the optimal coefficient of global compensation when joint-locked failure occurs at a particular time. Then considering the unexpectedness characteristics

of joint-locked failure, coefficient of global compensation vector is fitted in function form based on a priori failure information. In Section 4, simulations are implemented to validate the effectiveness of trajectory optimization strategy in reducing the velocity jump in both joint space and operation space when joint-locked failure occurs at arbitrary time. The last part is conclusion.

2. Mathematical Expression of Trajectory Optimization Strategy

Joints are the key parts of space manipulator operations. When joint-locked failure occurs, it will bring about velocity jumps in operation space and joint space, which would tremendously reduce the operability of space manipulator, especially the ability of pose-reaching and task-completing. In this section, trajectory optimization is proposed considering simultaneous velocity jumps reduction in two spaces. Based on the analysis of velocity jumps, the constraints and objective of trajectory optimization are derived.

Trajectory optimization is discussed on a typical space manipulator path planning task, which starts at time t_0 and ends at time t_e. The control interval is represented as Δt, and the execution time of path planning t can be represented as $t = t_0 + s\Delta t, 1 \le s \le (t_e - t_0)/\Delta t, s \in N^+$.

2.1. Analysis on Velocity Jumps in Operation Space and Joint Space. For a n degree space manipulator, when joints are healthy during path planning task, the Jacobian matrix and joint velocity vector at time t $(t_0 \le t \le t_e)$ can be expressed as

$$\mathbf{J}(\mathbf{q}(t)) = [\mathbf{j}_1(\mathbf{q}(t)), \ldots, \mathbf{j}_{k-1}(\mathbf{q}(t)), \mathbf{j}_k(\mathbf{q}(t)), \mathbf{j}_{k+1}(\mathbf{q}(t)),$$
$$\ldots, \mathbf{j}_n(\mathbf{q}(t))] \in \mathbf{R}^{r \times n},$$
$$\dot{\mathbf{q}}(t) = [\dot{q}_1(t), \ldots, \dot{q}_{k-1}(t), \dot{q}_k(t), \dot{q}_{k+1}(t), \ldots, \dot{q}_n(t)]^{\mathrm{T}}$$
$$\in \mathbf{R}^{n \times 1}, \quad t_0 \le t \le t_e. \tag{1}$$

The Jacobian matrix $\mathbf{J}$ is the function of joint angle vector $\mathbf{q}(t)$ at current time t. And r represents the dimension of operation space. $\mathbf{j}_i$ $(i \in [1, n])$ represents the ith column of Jacobian matrix.

When joint-locked failure occurs in joint k at time t_f, the dimension of joint space degenerates to $n - 1$. Correspondingly, the Jacobian matrix degenerates to $^k\mathbf{J}$, which is named as reduced Jacobian matrix [32]. The reduced Jacobian matrix and joint velocity vector at time t $(t_f < t \le t_e)$ can be expressed as

$$^k\mathbf{J}\left(^k\mathbf{q}(t)\right) = \left[^k\mathbf{j}_1\left(^k\mathbf{q}(t)\right), \ldots, ^k\mathbf{j}_{k-1}\left(^k\mathbf{q}(t)\right),\right.$$
$$\left.^k\mathbf{j}_{k+1}\left(^k\mathbf{q}(t)\right), \ldots, ^k\mathbf{j}_n\left(^k\mathbf{q}(t)\right)\right] \in \mathbf{R}^{r \times (n-1)},$$
$$^k\dot{\mathbf{q}}(t) = \left[^k\dot{q}_1(t), \ldots, ^k\dot{q}_{k-1}(t), ^k\dot{q}_{k+1}(t), \ldots, ^k\dot{q}_n(t)\right]^{\mathrm{T}}$$
$$\in \mathbf{R}^{(n-1) \times 1}, \quad t_f < t \le t_e. \tag{2}$$

Correspondingly, $^k\mathbf{j}_i$ $(i \in [1, n] \cap i \ne k)$ represents the ith column of reduced Jacobian matrix. And $^k\mathbf{q}(t)$ represents the reduced joint angle vector. After joint k fails at time t_f, the velocity of failure joints turns to 0, and velocity of healthy joints keeps constant. Thus, at time t $(t_f < t \le t_e)$, it has

$$\dot{q}_i(t) = {}^k\dot{q}_i(t),$$
$$\mathbf{j}_i(\mathbf{q}(t)) \ne {}^k\mathbf{j}_i\left({}^k\mathbf{q}(t)\right), \tag{3}$$
$$t_f < t \le t_e, \ i \in [1, n] \cap i \ne k.$$

Then the velocity jump vector in joint space at time t $(t_f < t \le t_e)$ can be expressed as

$$\Delta\dot{\mathbf{q}}(t) = [0 \ \cdots \ \dot{q}_k(t) \ \cdots \ 0]^{\mathrm{T}}. \tag{4}$$

$\dot{q}_k(t)$ represents the nominal velocity of kth joint when the joint is healthy. On the basis, the velocity jump in operation space $\Delta\dot{\mathbf{x}}(t)$ caused by joint-locked failure is expressed as

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{J}\dot{\mathbf{q}}(t) - {}^k\mathbf{J}^k\dot{\mathbf{q}}(t). \tag{5}$$

According to (5), it will lead to accumulative pose (position and orientation) deviation $\Delta\mathbf{P}$ in operation space:

$$\Delta\mathbf{P}(t) = \int_{t_f}^{t} \Delta\dot{\mathbf{x}}(t) \, dt = \int_{t_f}^{t} \left(\mathbf{J}\dot{\mathbf{q}}(t) - {}^k\mathbf{J}^k\dot{\mathbf{q}}(t)\right) dt. \tag{6}$$

When joint failure occurs unexpectedly during path planning task, an average velocity deviation in the operation space can be obtained:

$$\Delta\bar{\dot{\mathbf{x}}}(t) = \frac{\Delta\mathbf{P}(t)}{(t_e - t_f)} = \frac{1}{t_e - t_f} \int_{t_f}^{t_e} \left(\mathbf{J}\dot{\mathbf{q}}(t) - {}^k\mathbf{J}^k\dot{\mathbf{q}}(t)\right) dt. \tag{7}$$

Equation (7) can be used to evaluate the influence of velocity deviation caused by an unexpected joint-locked failure, while, for a general path planning task, the pose accuracy should fulfill accuracy threshold at each time to guarantee task completion. Set the threshold value as $\mathbf{l}$; when $|\Delta\mathbf{P}(t)| > |\mathbf{l}|$, the task cannot be completed. Therefore, the velocity jump in operation space should be eliminated preferentially to guarantee task completion.

Generally, velocity jump in operation space can be reduced by compensation for healthy joint. When compensation vector $\Delta^k\dot{\mathbf{q}}(t)$ $(t \in (t_f, t_e])$ is introduced, (5) is turned into

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{J}\dot{\mathbf{q}}(t) - {}^k\mathbf{J}\left({}^k\dot{\mathbf{q}}(t) + \Delta^k\dot{\mathbf{q}}(t)\right). \tag{8}$$

When the velocity deviation in operation space is completely eliminated, namely, $\Delta\dot{\mathbf{x}}(t) = \mathbf{0}$, the compensation vector for joint velocity can be expressed as

$$\Delta^k\dot{\mathbf{q}}(t) = \left({}^k\mathbf{J}\right)^{\dagger} \mathbf{J}\dot{\mathbf{q}}(t) - {}^k\dot{\mathbf{q}}(t). \tag{9}$$

$(^{k}\mathbf{J})^{\dagger}$ represents the pseudoinverse of the reduced Jacobian matrix. By given the compensation $\Delta^{k}\dot{\mathbf{q}}(t)$ as (9), the velocity jump in operation space can be completely eliminated. The contribution of failure joint is redistributed to healthy joints. However, additional velocity jumps are introduced into healthy joints. The velocity compensation $\Delta^{k}\dot{\mathbf{q}}(t)$ may be too large for safety threshold of joint velocity and the impact caused by compensation may damage the joint structure, leading to terribly concatenate failure of joints. Thus, it is of equal importance to decrease the velocity jumps in operation space and joint space.

2.2. Constraints and Objective of Trajectory Optimization. In order to achieve simultaneous velocity jumps reduction in both operation space and joint space, the velocity in operation space is considered as equality constraint, which is preferentially guaranteed for task completion. And joint velocity is used to construct the objective function, which is adjusted to achieve joint velocity jumps minimal. The constraints and objective function of trajectory optimization can be expressed as follows.

2.2.1. Constraints. Assume only position accuracy in operation space is considered during path planning task. The actual value of linear velocity in operation space should be in accordance with the nominal value; namely,

$$\mathbf{h}_{1}(t) = \mathbf{v}_{e}^{a}(t) - \mathbf{v}_{e}^{d}(t). \qquad (10)$$

$\mathbf{v}_{e}^{a}, \mathbf{v}_{e}^{d} \in \mathbf{R}^{3 \times 1}$ represent the actual and nominal value of the linear velocity in the operation space. Correspondingly, the position in operation space should be tracked as the nominal value:

$$\mathbf{h}_{2}(t) = \mathbf{P}_{e}^{a}(t) - \mathbf{P}_{e}^{d}(t) = \int_{t0}^{t} \mathbf{v}_{e}^{a}(t)\, dt - \int_{t0}^{t} \mathbf{v}_{e}^{d}(t)\, dt. \quad (11)$$

$\mathbf{P}_{e}^{a}, \mathbf{P}_{e}^{d} \in \mathbf{R}^{3 \times 1}$ represent the actual and nominal position in operation space, respectively. Equations (10) and (11) are named as task constraints. When task constraints are guaranteed, it has $\mathbf{h}_{i}(t) = \mathbf{0}^{3 \times 1}$ ($i = 1, 2$; $\mathbf{0}^{3 \times 1} \in \mathbf{R}^{3 \times 1}$).

Besides, the parameters of space manipulator should fulfill the thresholds in path planning. Constraints of joint velocity and joint angular acceleration are taken into account:

$$\mathbf{g}_{1,j}(t) = \left|\dot{q}_{j}(t)\right| - \dot{q}_{j}^{\max},$$

$$\mathbf{g}_{2,j}(t) = \left|\ddot{q}_{j}(t)\right| - \ddot{q}_{j}^{\max}$$

$$= \left|\frac{\left(\dot{q}_{j}(t + \Delta t) - \dot{q}_{j}(t)\right)}{\Delta t}\right| - \ddot{q}_{j}^{\max}, \qquad (12)$$

wherein $j = 1, \ldots, n$ represents the number of joints and $\dot{q}_{j}^{\max}$ and $\ddot{q}_{j}^{\max}$ represent the threshold value of joint velocity and angular acceleration of the jth joint, respectively. When (12) is fulfilled, it has $\mathbf{g}_{i,j}(t) \leq 0$ ($i = 1, 2$).

2.2.2. Objective Function. The objective function is constructed as the joint velocity deviation at failure time t_{f}. Since the dimension of joint velocity is reduced after joint failure, in order to guarantee the dimension, $^{k}\widetilde{\dot{\mathbf{q}}}$ is defined based on $^{k}\dot{\mathbf{q}}$:

$$^{k}\widetilde{\dot{\mathbf{q}}} = \left[^{k}\dot{q}_{1}, \ldots, ^{k}\dot{q}_{k-1}, 0, ^{k}\dot{q}_{k+1}, \ldots, ^{k}\dot{q}_{n}\right]^{\mathrm{T}} \in \mathbf{R}^{n \times 1}. \qquad (13)$$

Then the objective function can be expressed as

$$\mathbf{f} = \left\| \Delta^{k}\widetilde{\dot{\mathbf{q}}}\left(t_{f}\right) - \Delta\dot{\mathbf{q}}\left(t_{f} - \Delta t\right) \right\|. \qquad (14)$$

Based on the analysis above, the mathematical expression of trajectory optimization strategy is derived as

$$\begin{aligned} \min \quad & \mathbf{f} \\ \text{s.t.} \quad & \mathbf{g}_{i,j}(t) \leq 0 \quad (i = 1, 2; j = 1, \ldots, 7) \\ & \mathbf{h}_{k}(t) = \mathbf{0}^{3 \times 1} \quad \left(k = 1, 2; \mathbf{0}^{3 \times 1} \in \mathbf{R}^{3 \times 1}\right). \end{aligned} \qquad (15)$$

3. Solution for Trajectory Optimization considering Unexpectedness Characteristics of Joint-Locked Failure

In this section, trajectory optimization when joint fails at a particular time is solved firstly. A global compensation vector for joint velocity is constructed based on the orthogonal basis of Jacobian matrix null space. Then the optimal coefficient of compensation vector is obtained when the objective function is minimal. Since global compensation vector compensates joint velocity in each control interval ignoring the failure time, time delay problem can be efficiently avoided. Taking the failure time and corresponding optimal coefficient as a priori failure information, the law of optimal coefficient related to arbitrary failure time is concluded, and a method for coefficient function fitting is proposed based on a priori failure information to guarantee the unexpectedness characteristics of joint-locked failure.

3.1. Global Compensation Vector for Joint Velocity. In order to construct the global compensation vector, the orthogonal basis of Jacobian matrix null space $\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}$ is used to represent the self-motion characteristics of space manipulator. Manipulability [33] $w = \sqrt{\det(\mathbf{J}\mathbf{J}^{\mathrm{T}})}$ is introduced into the vector, whose gradient can be used to represent the velocity compensation weight of each joint. When joint-locked failure occurs, it is reduced as $^{k}w = \sqrt{\det(^{k}\mathbf{J}\,^{k}\mathbf{J}^{\mathrm{T}})}$. Besides, a constant coefficient c_{k} is used to adjust the velocity compensation rate for each joint. Then for each time, it has

$$\mathbf{u}(t) = c_{k}\left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\nabla w\left(\mathbf{q}(t)\right) \quad t_{0} \leq t < t_{f}, \qquad (16a)$$

$$^{k}\mathbf{u}(t) = c_{k}\left(\mathbf{I} - \left(^{k}\mathbf{J}\right)^{\dagger}\left(^{k}\mathbf{J}\right)\right)\nabla^{k}w\left(^{k}\mathbf{q}(t)\right)$$

$$\qquad\qquad\qquad\qquad t_{f} \leq t \leq t_{e}. \qquad (16b)$$

In (16a) and (16b), $\mathbf{u}(t) \in \mathbf{R}^{n \times 1}$ and $^k\mathbf{u}(t) \in \mathbf{R}^{(n-1) \times 1}$ are used to compensate joint velocity at each time in path planning. Before failure occurs, precompensation is implemented with the vector of (16a). After failure occurs, joint velocity is compensated according to (16b). Namely, joint velocity at each time t corresponds to a vector $\mathbf{u}(t)$ or $^k\mathbf{u}(t)$. Then the global compensation vector for joint velocity over the entire task is constructed:

$$\mathbf{U} = \left\{ \mathbf{u}(t_0), \mathbf{u}(t_0 + \Delta t), \ldots, \mathbf{u}(t_f - \Delta t), {}^k\mathbf{u}(t_f), \ldots, \right.$$

$$\left. {}^k\mathbf{u}(t_e) \right\}. \tag{17}$$

With the compensation of $\mathbf{U}$, joint velocity can be expressed as $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} + \mathbf{U}$, which is detailed in

$$\dot{\mathbf{q}}(t + \Delta t) = \mathbf{J}^\dagger \dot{\mathbf{x}}(t) + c_k \left(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J} \right) \nabla w(\mathbf{q}(t))$$

$$t_0 \leq t < t_f,$$

$$^k\dot{\mathbf{q}}(t + \Delta t) = \left({}^k\mathbf{J} \right)^\dagger \dot{\mathbf{x}}(t) \tag{18}$$

$$+ c_k \left(\mathbf{I} - \left({}^k\mathbf{J} \right)^\dagger \left({}^k\mathbf{J} \right) \right) \nabla {}^k w \left({}^k\mathbf{q}(t) \right)$$

$$t_f \leq t \leq t_e.$$

Joint angle at time $t + \Delta t$ can be obtained as

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \dot{\mathbf{q}}(t + \Delta t) \Delta t,$$

$$^k\mathbf{q}(t + \Delta t) = {}^k\mathbf{q}(t) + {}^k\dot{\mathbf{q}}(t + \Delta t) \Delta t. \tag{19}$$

By substituting $\mathbf{q}(t + \Delta t)$ and $^k\mathbf{q}(t + \Delta t)$ into (16a) and (16b), $\mathbf{u}(t + \Delta t)$ and $^k\mathbf{u}(t + \Delta t)$ can be obtained. The recurrence relation between $\mathbf{u}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ is generated, which is shown in Figure 1.

Since the dimension of velocity is reduced after joint failure, similar to (13), $^k\tilde{\mathbf{u}}(t)$ is defined as the same dimension as $\mathbf{u}(t)$, whose kth element equals 0. Then, according to (18), the objective function can be turned into

$$\mathbf{f}(\mathbf{U}) = \left\| {}^k\mathbf{J}^\dagger \dot{\mathbf{x}}(t_f) - \mathbf{J}\dot{\mathbf{x}}(t_f - \Delta t) + {}^k\tilde{\mathbf{u}}(t_f) \right.$$

$$\left. - \mathbf{u}(t_f - \Delta t) \right\|. \tag{20}$$

For a particular path planning task, when failure time is certain, the global compensation vector $\mathbf{U}$ will change with coefficient c_k. Thus the objective function can be optimized to achieve joint velocity jump minimal by finding the optimal c_k.

3.2. Calculation for the Optimal Coefficient c_k.
In order to find the optimal c_k, the gradient of reduced manipulability in global compensation vector should be calculated firstly. It has

$$\nabla {}^k w \left({}^k\mathbf{q} \right) = \frac{\partial {}^k w}{\partial {}^k\mathbf{q}} = \left(\frac{\partial {}^k w}{\partial q_1}, \ldots, \frac{\partial {}^k w}{\partial q_i}, \ldots, \frac{\partial {}^k w}{\partial q_n} \right)^{\mathrm{T}} \tag{21}$$

$$i = 1, \ldots, n, \ i \neq k,$$

wherein the partial derivative of the reduced manipulability to the ith joint angle can be expressed as

$$\frac{\partial {}^k w}{\partial q_i} = \frac{\partial \sqrt{\det \left({}^k\mathbf{J} {}^k\mathbf{J}^{\mathrm{T}} \right)}}{\partial q_i} = \frac{\partial \left(\det \left({}^k\mathbf{J} {}^k\mathbf{J}^{\mathrm{T}} \right) \right) / \partial q_i}{2 \cdot {}^k w}. \tag{22}$$

Make $^k\mathbf{Jp} = {}^k\mathbf{J} {}^k\mathbf{J}^{\mathrm{T}}$, and $^k\mathbf{Jp}_i$ expresses the ith row of $^k\mathbf{Jp}$. Then

$$\frac{\partial \left(\det \left({}^k\mathbf{Jp} \right) \right)}{\partial q_i} = \sum_{j=1}^{n} \det \left({}^k\mathbf{Jp}_1, \ldots, \frac{\partial {}^k\mathbf{Jp}_j}{\partial q_i}, \ldots, {}^k\mathbf{Jp}_n \right), \tag{23}$$

wherein $\partial {}^k\mathbf{Jp}_j / \partial q_i = (\partial {}^k\mathbf{Jp} / \partial q_i)_j$ represents the jth row of $\partial {}^k\mathbf{Jp} / \partial q_i \in \mathbf{R}^{r \times r}$ and r is the dimension of operation space. Then

$$\frac{\partial {}^k\mathbf{Jp}}{\partial q_i} = \frac{\partial \left({}^k\mathbf{J} {}^k\mathbf{J}^{\mathrm{T}} \right)}{\partial q_i} = \frac{\partial {}^k\mathbf{J}}{\partial q_i} {}^k\mathbf{J}^{\mathrm{T}} + {}^k\mathbf{J} \left(\frac{\partial {}^k\mathbf{J}}{\partial q_i} \right)^{\mathrm{T}}. \tag{24}$$

In (24), $\partial {}^k\mathbf{J} / \partial q_i \in \mathbf{R}^{(n-1) \times r}$. Substitute (24) into (23); the partial derivative of reduced manipulability to joint angle q_i can be obtained:

$$\frac{\partial {}^k w}{\partial q_i} = \frac{\sum_{j=1}^{n} \det \left({}^k\mathbf{Jp}_1, \ldots, \left(\left(\partial {}^k\mathbf{J} / \partial q_i \right) {}^k\mathbf{J}^{\mathrm{T}} + {}^k\mathbf{J} \left(\partial {}^k\mathbf{J} / \partial q_i \right)^{\mathrm{T}} \right)_j, \ldots, {}^k\mathbf{Jp}_n \right)}{2 \cdot {}^k w}. \tag{25}$$

Substitute (25) into (21); the gradient of the reduced manipulability can be obtained. The gradient of manipulability can

be obtained in a similar way. Then, by given c_k, the value of global compensation vector $\mathbf{U}$ can be calculated.
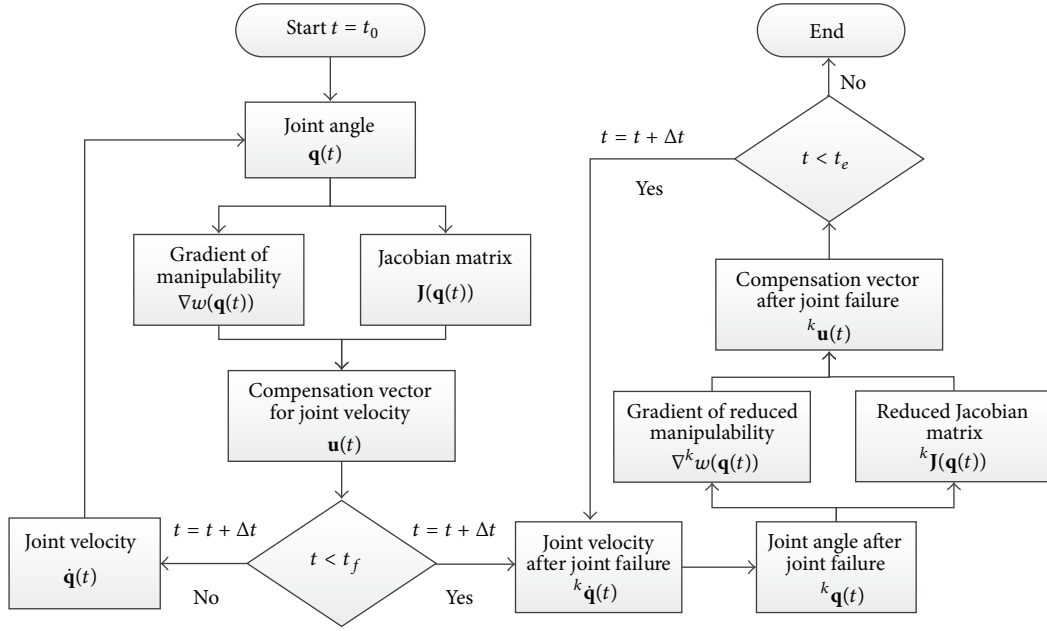
FIGURE 1: The recurrence relation between $\mathbf{u}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$.

Before searching for the optimal c_k, its feasible region should be preferentially determined. According to constraints (10) and (11), the feasible region for c_k can be defined as

$$c_k \in \left\{ c_k \mid \left(\mathbf{P}_e^a \left(\mathbf{U} \left(k_c \right) \right) - \mathbf{P}_e^d \right)_x \le l_x \right.$$

$$\cap \left(\mathbf{P}_e^a \left(\mathbf{U} \left(k_c \right) \right) - \mathbf{P}_e^d \right)_y \le l_y \cap \left(\mathbf{P}_e^a \left(\mathbf{U} \left(k_c \right) \right) - \mathbf{P}_e^d \right)_z \quad (26)$$

$$\left. \le l_z \right\},$$

wherein the subscripts x, y, z represent the x, y, z component of vector $(\mathbf{P}_e^a(\mathbf{U}(c_k)) - \mathbf{P}_e^d)$.

Equation (23) means the x, y, z component of deviation between actual position and desired position should fulfill the threshold value of position accuracy $\mathbf{l}$ simultaneously, where $\mathbf{l}$ is expressed as $\mathbf{l} = [l_x, l_y, l_z]^{\mathrm{T}}$. In the feasible region, the optimal c_k can be obtained as follows:

(a) According to the constraints, obtain the feasible region of $c_k \in [c_k^{\min}, c_k^{\max}]$.

(b) Set the initial value of $c_k = c_k^{\min}$ and interval value $\Delta c_k > 0$, and calculate the global compensation vector $\mathbf{U}(c_k)$.

(c) Calculate the joint velocity and acceleration compensated with $\mathbf{U}(c_k)$.

(d) Confirm whether joint velocity and acceleration guarantee the constraints (12). If the constraints are not guaranteed, the corresponding c_k is abandoned. Else the value of objective function is obtained $\mathbf{f}(c_k)$. $(c_k, \mathbf{f}(c_k))$ is stored in set $\mathbf{K}$.

(e) $c_k = c_k + \Delta c_k$. If $c_k \le c_k^{\max}$, turn to step (c). Else select the minimal value of $\mathbf{f}(c_k)$ in set $\mathbf{K}$, and the corresponding c_k is the optimal value.

The steps are also detailed as shown in Figure 2.

3.3. The Method of Fitting Optimal Coefficient Function. With the analysis above, the optimal c_k can be obtained to minimize the objective function. However, during practical application, the accurate failure time is not easy to be obtained. In this section, the trajectory optimization is developed considering the unexpectedness characteristics of joint-locked failure when failure information is unknown. And a method is proposed to derive the law of optimal coefficient for unexpected joint failure.

Since failure time is unexpected in practical application, the objective function is changed as the maximal value of joint velocity deviation between adjacent control cycles:
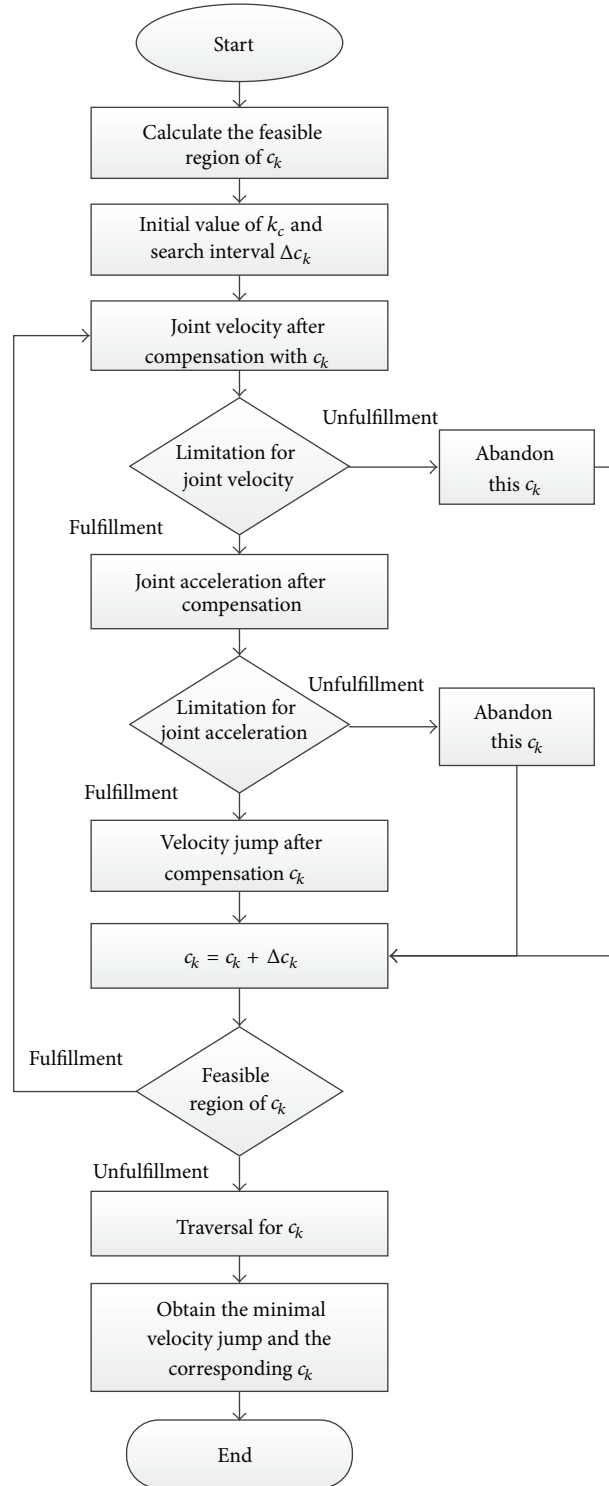
$$\mathbf{f} = \max_t \| \Delta \dot{\mathbf{q}} \left(t \right) \|,$$

$$t = \left\{ t_0 + \Delta t, t_0 + 2\Delta t, \dots, t_e - \Delta t, t_e \right\}, \quad (27)$$

wherein $\Delta \dot{\mathbf{q}}(t)$ represents the joint velocity deviation, which is expressed as

$$\Delta \dot{\mathbf{q}} \left(t \right) = \begin{cases} \dot{\mathbf{q}} \left(t \right) - \dot{\mathbf{q}} \left(t - \Delta t \right), & t < t_f \\ {}^k \tilde{\dot{\mathbf{q}}} \left(t \right) - \dot{\mathbf{q}} \left(t - \Delta t \right), & t = t_f \\ {}^k \tilde{\dot{\mathbf{q}}} \left(t \right) - {}^k \tilde{\dot{\mathbf{q}}} \left(t - \Delta t \right), & t > t_f. \end{cases} \quad (28)$$

According to the analysis in Sections 3.1 and 3.2, a unique c_k can be obtained to minimize the objective function when joint-locked failure occurs at particular time t_f. Thus, when

FIGURE 2: Process of finding the optimal c_k when joint fails at a particular time.

the failure time is unknown, a function of operation time $\mathbf{c}_k(t)$ can be constructed to guarantee the arbitrary failure time. Then (18) can be turned into

$$\dot{\mathbf{q}}(t + \Delta t) = \mathbf{J}^\dagger \dot{\mathbf{x}}(t) + \mathbf{c}_k(t)\left(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}\right) \nabla w(\mathbf{q}(t))$$

$$t_0 \leq t < t_f,$$

$${}^k\dot{\mathbf{q}}(t + \Delta t) = \left({}^k\mathbf{J}\right)^\dagger \dot{\mathbf{x}}(t)$$

$$+ \mathbf{c}_k(t)\left(\mathbf{I} - \left({}^k\mathbf{J}\right)^\dagger \left({}^k\mathbf{J}\right)\right) \nabla {}^k w\left({}^k\mathbf{q}(t)\right)$$

$$t_f \leq t \leq t_e.$$

$$(29)$$

As a basis, the trajectory optimization strategy considering the unexpectedness characteristics of joint-locked failure can be expressed as

$$
\begin{aligned}
\text{find} \quad & \text{function } \mathbf{c}_k(t) \\
\text{min} \quad & \mathbf{f} = \max_t \|\Delta \dot{\mathbf{q}}(t)\| \\
& t = \{t_0 + \Delta t, t_0 + 2\Delta t, \ldots, t_0 + s\Delta t, \ldots, t_e\} \quad (30) \\
\text{s.t.} \quad & \mathbf{g}_{i,j}(t) \leq 0 \quad (i = 1, 2; j = 1, \ldots, 7) \\
& \mathbf{h}_k(t) = \mathbf{0}^{3 \times 1} \quad \left(k = 1, 2; \mathbf{0}^{3 \times 1} \in \mathbf{R}^{3 \times 1}\right).
\end{aligned}
$$

In order to solve the optimization strategy and obtain the coefficient function $\mathbf{c}_k(t)$, a method of fitting the coefficient function based on polynomial fitting using a priori failure information is proposed. Define the polynomial function as follows:

$$
\mathbf{c}_k(t) = \sum_{i=0}^{s} a_i \cdot t^i. \tag{31}
$$

$[a_0, a_1, \ldots, a_s]^{\mathrm{T}} \in \mathbf{R}^{(s+1) \times 1}$ represents the coefficient of polynomial, s represents the order of polynomial, and t represents the task execution time. Generally, joint velocity at the initial time t_0 and the end time t_e should be zero during path planning; namely, $\dot{\mathbf{q}}(t_0) = \mathbf{0}$ and $\dot{\mathbf{q}}(t_e) = \mathbf{0}$. According to $\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \mathbf{U}$, function $\mathbf{c}_k(t)$ should guarantee the boundary condition:

$$
\begin{aligned}
\mathbf{c}_k(t_0) &= 0, \\
\mathbf{c}_k(t_e) &= 0.
\end{aligned} \tag{32}
$$

In order to obtain the polynomial coefficient $[a_0, a_1, \ldots, a_s]^{\mathrm{T}} \in \mathbf{R}^{(s+1) \times 1}$, a priori failure information is used to fit function (31). The a priori failure information includes failure time, failure joint angle, and the corresponding optimal coefficient c_k. When failure time is certain, the failure joint angle can be theoretically calculated via manipulator path planning algorithm, and the optimal c_k can be obtained based on the analysis in Section 3.2. In this way, a priori failure information can be obtained by supposing possible failure time in path planning. Then numbers of failure time $\mathbf{t}^{\mathrm{sam}} = (t_{f1}, t_{f2}, \ldots, t_{fm})$ and the corresponding optimal coefficient $\mathbf{c}_k^{\mathrm{sam}} = (c_{k1}, c_{k2}, \ldots, c_{km})$ can be obtained as samples for function fitting.

The samples consist of failure time $\mathbf{t}^{\mathrm{sam}} = (t_{f1}, t_{f2}, \ldots, t_{fm})$ and the corresponding optimal coefficient $\mathbf{c}_k^{\mathrm{sam}} = (c_{k1}, c_{k2}, \ldots, c_{km})$. The ith sample can be represented as (t_{fi}, c_{ki}). t_{fi} and c_{ki} are the ith element of $\mathbf{t}^{\mathrm{sam}}$ and $\mathbf{c}_k^{\mathrm{sam}}$, which can also be represented as $(\mathbf{t}^{\mathrm{sam}})_i$ and $(\mathbf{c}_k^{\mathrm{sam}})_i$, respectively. Thus, the ith sample can be represented as $((\mathbf{t}^{\mathrm{sam}})_i, (\mathbf{c}_k^{\mathrm{sam}})_i)$.

Define the evaluation criterion of function fitting e as the sum of square for deviations between samples and fitting function value, which has the expression of

$$
e = \sum_{i=1}^{m} \left| (\mathbf{c}_k^{\mathrm{sam}})_i - \mathbf{c}_k((\mathbf{t}^{\mathrm{sam}})_i) \right|^2. \tag{33}
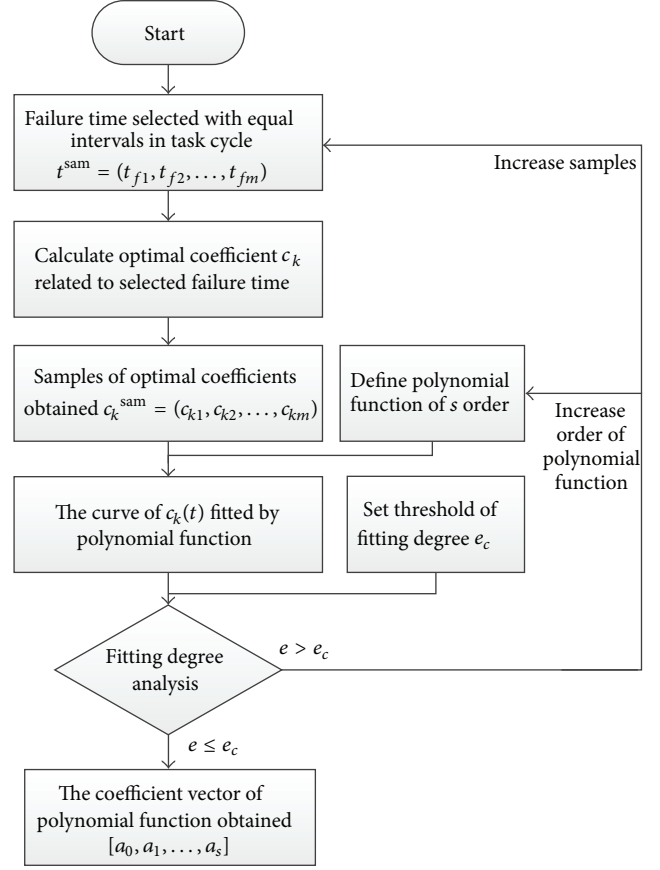$$

FIGURE 3: Calculation process of function $\mathbf{c}_k(t)$.

The threshold of evaluation criterion is defined as e_c. When $e \leq e_c$, function $\mathbf{c}_k(t)$ has a high fitting quality, and the fitting result is acceptable, while $e > e_c$ means more samples should be selected or polynomial order should be increased for better fitting. In conclusion, the process of fitting function $\mathbf{c}_k(t)$ is shown in Figure 3.

By considering the unexpectedness characteristics of joint-locked failure, a method is proposed based on a priori failure information for optimal coefficient function fitting. So when accurate joint failure information cannot be acquired in practical application, global compensation vector based on $\mathbf{c}_k(t)$ can be introduced to achieve trajectory optimization, with which velocity jumps can be efficiently reduced no matter when joint-locked failure occurs.

4. Simulation Experiments

4.1. Research Object. An SSRMS [2] type space manipulator who has 7 DOF is used to carry out simulation experiments. The coordinate systems are shown in Figure 4, and the DH parameters are listed in Table 1. Path planning task mentioned in the paper concerns the position accuracy in the operation space of space manipulator only.

4.2. Velocity Jump Reduction When Joint-Locked Failure Occurs at a Particular Time. In a particular path planning
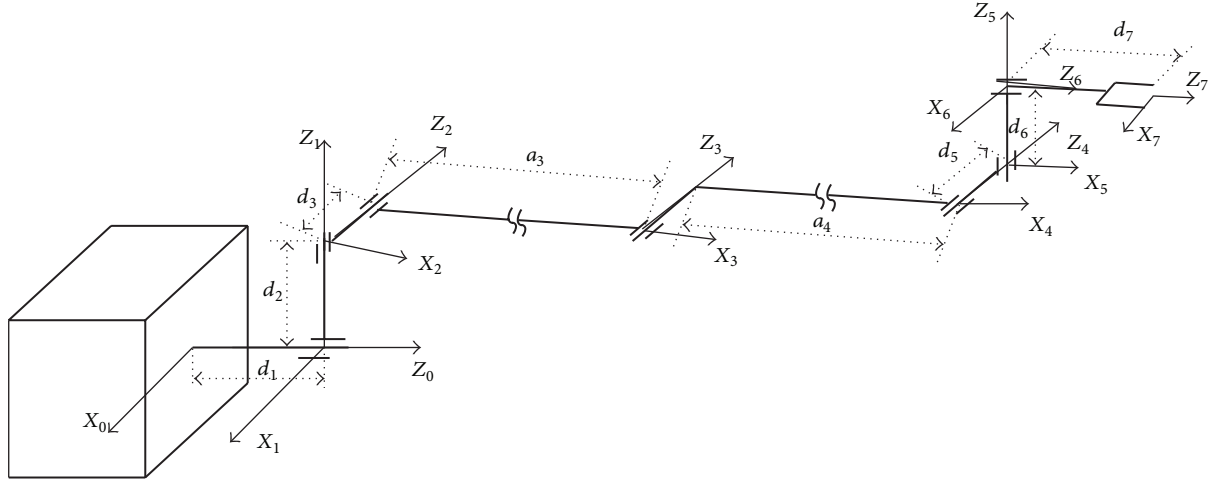
FIGURE 4: Coordinate systems of 7 DOF space manipulator.

TABLE 1: DH parameters of space manipulator.

i	θ_i (°)	d_i (m)	a_{i-1} (m)	α_{i-1} (°)
1	0	0.6	0	90
2	90	0.5	0	−90
3	0	0.5	5	0
4	0	0.5	5	0
5	0	0.5	0	90
6	−90	0.5	0	−90
7	0	0.6	0	0

task, the initial configuration of space manipulator is set as $[-50, -170, 150, -60, 130, 170, 0]$(°) and the target pose is set as $[9.6, 0, 3, -1, -0.5, -2]$. The other parameters of path planning task are listed in Table 2.

According to Sections 3.1 and 3.2, under the position threshold of $\mathbf{l}$, the feasible region of c_k is calculated as $c_k \in [0, 0.36]$. Then set the initial value of c_k as 0, and carry out search algorithm with interval $\Delta c_k = 0.01$ in the feasible region. The value of objective function $\mathbf{f}$ changing with c_k is shown in Figure 5. When $c_k = 0.25$, the objective function is minimal; $\mathbf{f}_{\min} = 0.0035$ rad/s.

When $c_k = 0.25$, the joint velocities are shown in Figure 6. Figure 6(a) represents the velocity curve of joint 2 (failure joint), and Figure 6(b) represents the velocity curve of joint 3 (healthy joint). It is obvious that the joint velocity jump is significantly reduced after global compensation. In detail, the value of velocity jumps, acceleration, and jump reduction rate of each joint is listed in Table 3.

In Table 3, velocity jump of each joint is reduced significantly except joint 5. Since objective function reflects the accumulation of velocity jump of all the joints, when objective function is minimal, it does not mean the velocity jump of each joint is minimal. The velocity jumps of some joints may increase, such as joint 5 in Table 3. In a former paper of the author [28], the coefficient is changed into diagonal matrix form to achieve velocity jump minimal for each joint. However, it brings about higher computation cost and more
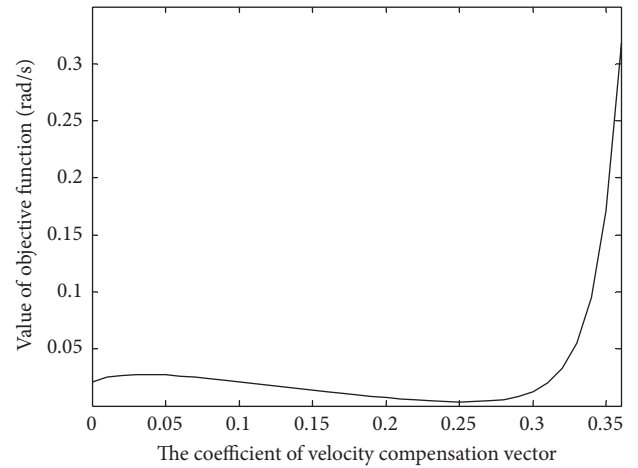
FIGURE 5: Value of objective function changes with c_k.

complex solution process. With constant coefficient of c_k, the norm of velocity jumps is reduced 84.55%, which shows great efficiency of the global compensation vector constructed with constant coefficient in reducing joint velocity jumps. Although jumps of some joints (joint 5) increase, the velocity jump is very small and can be accepted in practical application.

Besides, the position deviation in operation space when $c_k = 0.25$ is obtained in Figure 7. With the compensation in joint velocity at each control interval, the position deviation in operation space is accumulated. The maximum deviation is less than 4×10^{-3} m, which is much smaller than the length of manipulator (10^1 order of magnitude). During the 20 s' path planning task, the velocity jump in operation space is about 10^{-4} m/s order of magnitude, which can be basically ignored. In conclusion, the velocity jumps in operation space can be mostly eliminated, while the velocity jump in joint space can be reduced to minimal when joint-locked failure occurs at a particular time.

TABLE 2: Parameters for path planning task.

Task cycle	Control interval	Failure joint	Failure time	Position threshold
20 s	0.05 s	2nd joint	10 s	$\mathbf{l}$ = (0.01 m, 0.01 m, 0.01 m)

TABLE 3: Analysis of velocity jump and acceleration with compensation for each joint.

Joint i	Velocity jump before compensation (rad/s)	Acceleration before compensation (rad/s^2)	Velocity jump after compensation (rad/s)	Acceleration after compensation (rad/s^2)	Reduction rate of velocity jump
1	7.891×10^{-3}	1.578×10^{-1}	1.476×10^{-3}	2.952×10^{-2}	81.29%
2	1.940×10^{-2}	3.880×10^{-1}	2.584×10^{-3}	5.168×10^{-2}	86.68%
3	3.682×10^{-3}	7.364×10^{-2}	1.185×10^{-3}	2.370×10^{-2}	67.82%
4	5.114×10^{-4}	1.023×10^{-2}	3.323×10^{-4}	6.646×10^{-3}	35.02%
5	2.774×10^{-4}	5.548×10^{-3}	6.454×10^{-4}	1.291×10^{-2}	−57.62%
6	6.048×10^{-4}	1.210×10^{-2}	1.259×10^{-4}	2.518×10^{-3}	79.18%
7	0	0	0	0	0
Objective function	2.128×10^{-2}	4.256×10^{-1}	3.287×10^{-3}	5.673×10^{-2}	84.55%

(a) Joint 2

(b) Joint 3
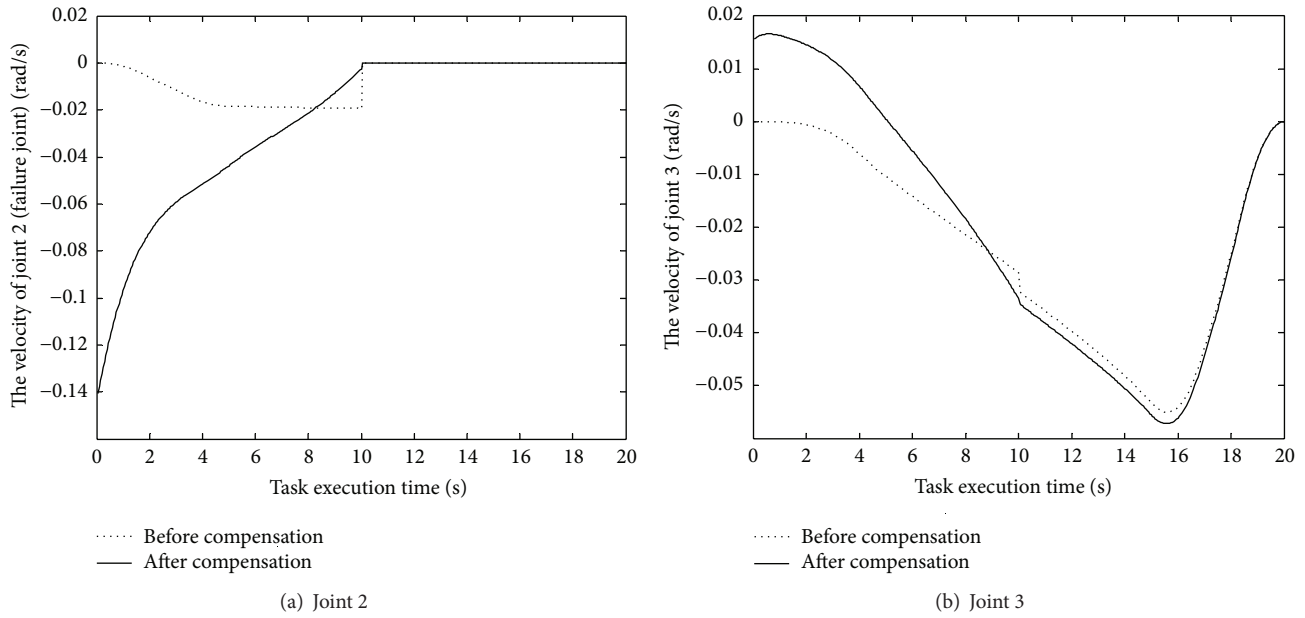
FIGURE 6: Joint velocity compensated when $c_k = 0.25$.

4.3. Coefficient Function Fitting for Unexpected Joint-Locked Failure.

Define the initial configuration of space manipulator path planning task as [−61.5146   −176.3220   145.8516   −60.7288   129.9836   169.1376   0](°), and set the target pose as [9.6 m, 0 m, 3 m, −1 rad, −0.5 rad, −2 rad]. The velocity in operation space is planned by trapezoidal planning with parabola transition. Task cycle is 12 s, acceleration time is 2 s, and control interval is 50 ms. Joint-locked failure is assumed to occur in joint 2.

Since joint-locked failure time is unexpected, a priori failure information is used to fit the coefficient function. A priori failure information includes possible failure time, failure joint angle, and the corresponding optimal c_k. During the mentioned path planning task, 40 moments are selected
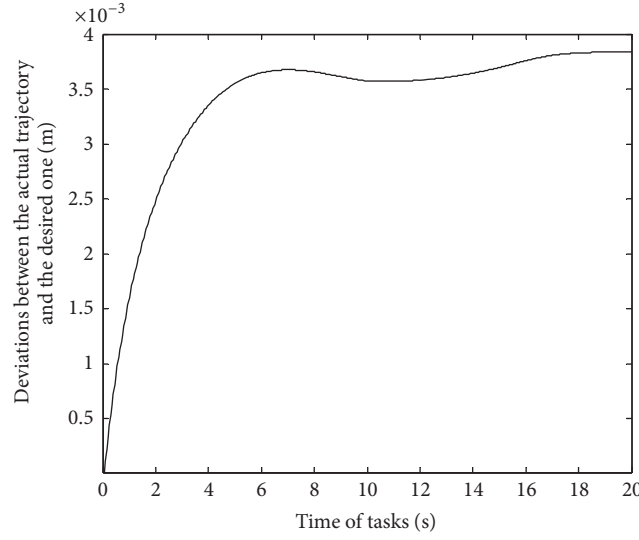
FIGURE 7: Position deviations in operation space after velocity reduction.

uniformly as possible failure time. According to (18) and (19), the joint angle at failure time can be obtained. Correspondingly, the optimal c_k related to each moment can be obtained, which is shown in Figure 8. Regardless of the samples $c_k = 0$, the other samples can be used to fit a five-order polynomial function when meeting the threshold of fitting $e_c = 10^{-4}$:

$$\mathbf{c}(t) = -0.0005t^5 + 0.0175t^4 - 0.2341t^3 + 1.532t^2 \\ - 5.0034t + 6.9496. \tag{34}$$

Considering the samples $c_k = 0$, the coefficient function for an unexpected joint-locked failure can be expressed as

$$\mathbf{c}_k(t) = \begin{cases} 0 & 0 \le t < t_{s1} \\ \mathbf{c}(t) & t_{s1} \le t \le t_{s2} \\ 0 & t_{s2} \le t < t_e. \end{cases} \tag{35}$$

t_{s1} and t_{s2} are segment points. The criterion to determine the segment points is defined as follows: the values of objective function compensated with $c_k = \mathbf{c}_k(t_s)$ and $c_k = 0$ are the same; namely,

$$\left| \mathbf{f}_{c_k = \mathbf{c}_k(t_s)} - \mathbf{f}_{c_k = 0} \right| \in \varepsilon. \tag{36}$$

ε represents infinitesimal. However, for each possible failure moment t, there is a feasible region for the related coefficient $^t c_k \in [^t c_k{}^{\min}, {}^t c_k{}^{\max}]$. If t_s selected according to (36) guarantees $\mathbf{c}(t_s) \notin [^{t_s} c_k{}^{\min}, {}^{t_s} c_k{}^{\max}]$, it should be abandoned. Instead of it, the segment points are set as $t_s = \{t_s \mid \mathbf{c}(t) = {}^{t_s} c_k{}^{\max}$ or $\mathbf{c}(t) = {}^{t_s} c_k{}^{\min}, t_s \in R^+\}$.

Accordingly, two segment points can be obtained as $t_{s1} = 1.1\,\text{s}, t_{s2} = 10.9\,\text{s}$. Then (35) can be expressed as

$$\mathbf{c}_k(t) = \begin{cases} 0 & 0 \le t < 1.1 \\ -0.0005t^5 + 0.0175t^4 - 0.2341t^3 + 1.532t^2 - 5.0034t + 6.9496 & 1.1 \le t \le 10.9 \\ 0 & 10.9 \le t < 12. \end{cases} \tag{37}$$

Based on (37), the compensation vector of each control interval $\mathbf{u}(\mathbf{c}_k(t))$ can be obtained, with which the velocity jump can be reduced no matter when joint-locked failure occurs.

4.4. Verification of the Coefficient Function Related to Unexpected Joint-Locked Failure. In order to verify the efficiency of the optimal coefficient function (37), assume joint-locked failure occurs at each control interval; namely, failure occurs at $0.05\,\text{s}, 0.1\,\text{s}, 0.15\,\text{s}, \ldots, 12\,\text{s}$. Via the velocity compensation

vector based on (37), the velocity jumps related to each failure time are shown in Figure 9.

At most failure time, the joint velocity jump can be significantly reduced. The average velocity jump equals $4.9e^{-3}\,\text{rad/s}$ after compensation. Compared with the average value of $0.0172\,\text{rad/s}$ before compensation, velocity jump is reduced 71.5% averagely. Especially in period of $t_{s3}-t_{s4}$, the performance of jump reduction is remarkable, and the velocity jump is reduced below $5e^{-3}\,\text{rad/s}$.
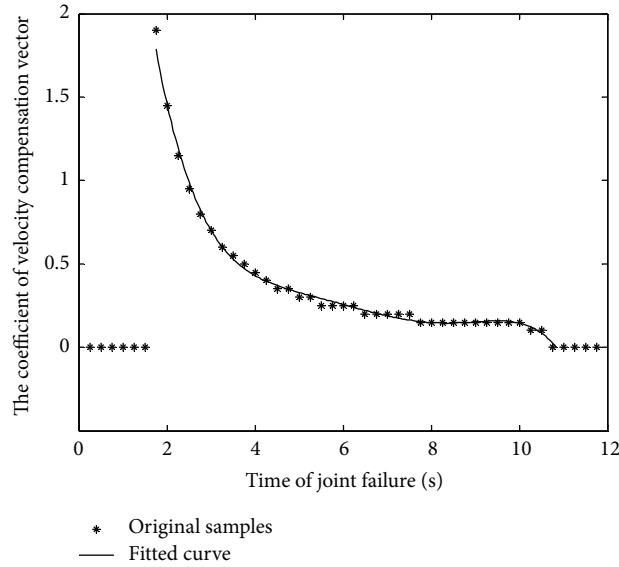
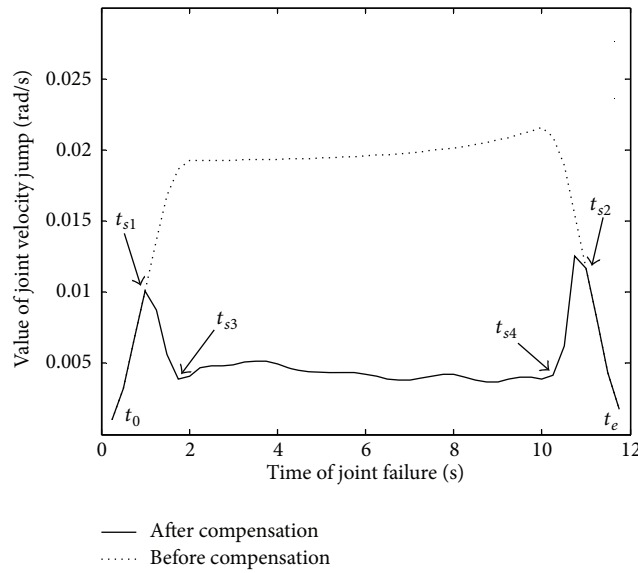FIGURE 8: Samples and fitted polynomial function of $\mathbf{c}_k(t)$.

FIGURE 9: Joint velocity jump comparison along the task cycle.

Before velocity jump reduction, during a control interval $\Delta t = 0.05$ s, joint acceleration needs to change 0.344 rad/s² averagely when joint-locked failure occurs. In order to afford the acceleration, huge joint torque is needed. When the safe threshold of joint acceleration is set as 0.15 rad/s² to protect joint torque not exceeding the rated torque, the change of velocity cannot be achieved and huge impact will be affected on joint motor. After velocity jump reduction, the requirement for joint acceleration is reduced to 0.098 rad/s² averagely, which fulfills the safe threshold and the impact on the joint can be eliminated. In this way, the proposed trajectory strategy is of great efficiency in smoothing the velocity parameter and guaranteeing motion safety.

During periods of 0–t_{s1} and t_{s2}–t_e, the reduction performance of velocity jump is worse than other periods. According to (37), the compensation coefficient in this period equals 0, which means no compensation is introduced. So when joint-locked failure occurs in this period, the velocity jump cannot be reduced with the proposed method. However, the velocity jump in this period is below 0.01 rad/s (0.57°/s), which is acceptable for space manipulator in most applications.

Periods of t_0–t_{s3} and t_{s4}–t_e are almost consistent with the acceleration stage and deceleration stage of path planning. Velocity in operation space changes quickly in the two stages, making the velocity compensation insufficient for jump

reduction. As a result, the reduction performance is not ideal in these periods. Period t_{s3}–t_{s4} corresponds to the uniform motion stage of path planning, the velocity in operation space during this stage is stable, and the velocity jump reduction is significant. It can be concluded that velocity changing in operation space at the failure time can influence the performance of velocity jump reduction.

In conclusion, the coefficient function obtained based on a priori failure information is verified efficiently in reducing the velocity jumps caused by an unexpected joint-locked failure. And velocity changing in operation space at the failure time can influence the reduction performance. The limitation of the method is that when joint-locked failure occurs at the initial and end period of path planning, the velocity jump cannot be reduced. However, velocity jumps in these periods are always small, which can be ignored in space manipulator applications.

5. Conclusion

With the proposed trajectory optimization strategy in the paper, the on-orbit operations can be executed to completion without shutting down the space manipulator when joint-locked failure occurs unexpectedly. The velocity jumps in both joint space and operation space caused by joint-locked failure are simultaneously reduced, and the impact caused on space manipulator structure can be eliminated. As a result, the safety of space manipulator and related parts can be guaranteed. The core of the strategy is the construction of global compensation vector for joint velocity, which consists of the gradient of reduced manipulability, orthogonal matrix of null space, and coefficient. The strategy can deal with velocity jumps reduction problem caused by joint-locked failure at either a particular time or an unexpected time. For joint-locked failure at a particular time, the optimal coefficient of compensation vector can be obtained to achieve velocity jump minimal. For joint-locked failure at an unexpected time, a priori failure information is obtained by assuming the possible failure time and calculating the corresponding optimal coefficient. Then the law of coefficient is concluded for trajectory optimization. As a result, coefficient is turned into function form and a function fitting method based on a priori failure information is proposed. With simulations, the efficiency of the method is verified in reducing the velocity jumps no matter when joint-locked failure occurs.

For further study, the effect of the initial configuration of manipulator on the velocity jump reduction performance should be discussed in-depth. Besides, the velocity in operation space is planned by the trapezoidal planning with parabola transition in the paper. However, the reduction performance in the acceleration and deceleration stage is not ideal compared with the uniform motion stage, which is considered to be the limitation of the velocity planning method in operation space. By changing the planning way of velocity in operation space, the reduction performance for velocity jump is worthy to be researched. In addition, the paper concentrates on space manipulator and single joint failure. For arbitrary serial manipulator, the trajectory

optimization strategy is adaptable, which can also be developed for multiple joint failure tolerance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] P. Putz, "Space robotics in Europe: a survey," *Robotics and Autonomous Systems*, vol. 23, no. 1-2, pp. 3–16, 1998.

[2] R. McGregor and L. Oshinowo, "Flight 6A: deployment and checkout of the space station remote manipulator system (SSRMS)," in *Proceedings of the 6th International Symposium Artificial Intelligence and Robotics & Automation in Space (i-SAIRAS '01)*, Saint-Hubert, Canada, June 2001.

[3] B. S. Dhillon, A. R. M. Fashandi, and K. L. Liu, "Robot systems reliability and safety: a review," *Journal of Quality in Maintenance Engineering*, vol. 8, no. 3, pp. 170–212, 2002.

[4] H. Abdi and S. Nahavandi, "Optimal actuator fault tolerance for static nonlinear systems based on minimum output velocity jump," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '10)*, pp. 1165–1170, Harbin, China, June 2010.

[5] S. ShiCai, W. Da, R. SiPu et al., "High integrated modular joint for Chinese Space Station Experiment Module Manipulator," in *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics (AIM '15)*, pp. 1195–1200, Busan, South Korea, July 2015.

[6] A. Rovetta, "'FRIEND' robot, space telerobot for rescue and recovery of astronauts in space stations," in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '91)*, vol. 3, pp. 1663–1668, IEEE, Osaka, Japan, 1991.

[7] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure-tolerant control of kinematically redundant manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '97)*, vol. 3, pp. 2595–2600, April 1997.

[8] J. D. English and A. A. Maciejewski, "Failure tolerance through active braking: a kinematic approach," *The International Journal of Robotics Research*, vol. 20, no. 4, pp. 287–299, 2001.

[9] T. Yüksel and A. Sezgin, "A model-based fault function approximation scheme for robot manipulators using M-ANFIS," in *Proceedings of the 15th IEEE Mediterranean Electrotechnical Conference (MELECON '10)*, pp. 47–51, IEEE, Valletta, Malta, April 2010.

[10] S. J. Yoo, "Actuator fault detection and adaptive accommodation control of flexible-joint robots," *IET Control Theory & Applications*, vol. 6, no. 10, pp. 1497–1507, 2012.

[11] L. M. Capisani, A. Ferrara, A. Ferreira de Loza, and L. M. Fridman, "Manipulator fault diagnosis via higher order sliding-mode observers," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 10, pp. 3979–3986, 2012.

[12] L. M. Capisani, A. Ferrara, and P. Pisu, "Sliding mode observers for vision-based fault detection, isolation and identification in robot manipulators," in *Proceedings of the American Control Conference (ACC '10)*, pp. 4540–4545, IEEE, Baltimore, Md, USA, July 2010.

[13] M. Goel, A. A. Maciejewski, and V. Balakrishnan, "Undetected locked-joint failures in kinematically redundant manipulators: a workspace analysis," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 317–322, IEEE, Vancouver, Canada, October 1998.

[14] R. G. Roberts, R. S. Jamisola Jr., and A. A. Maciejewski, "Identifying the failure-tolerant workspace boundaries of a kinematically redundant manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 4517–4523, IEEE, Roma, Italy, April 2007.

[15] C. L. Lewis and A. A. Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of joint failures," *Computers & Electrical Engineering*, vol. 20, no. 3, pp. 273–288, 1994.

[16] J. Li, J. Li, Z. Wu, and Q. Zhang, "Fault tolerance of redundant manipulators with multi-joint failed and its optimization," *Chinese Journal of Mechanical Engineering*, vol. 38, no. 7, pp. 111–115, 2002.

[17] Y. Chen, J. E. McInroy, and Y. Yi, "Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 680–691, 2003.

[18] 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.

[19] K. Li and Y. Zhang, "Fault-tolerant motion planning and control of redundant manipulator," *Control Engineering Practice*, vol. 20, no. 3, pp. 282–292, 2012.

[20] J. Zhao and D. Feng, "Comprehensive evaluation of fault-tolerant properties of redundant robots," *Chinese Journal of Mechanical Engineering*, vol. 21, no. 4, pp. 22–26, 2008.

[21] Q. Li and J. Zhao, "A universal approach for configuration synthesis of reconfigurable robots based on fault tolerant indices," *Industrial Robot*, vol. 39, no. 1, pp. 69–78, 2012.

[22] S. K. Ralph and D. K. Pai, "Computing fault tolerant motions for a robot manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '99)*, vol. 1, pp. 486–493, IEEE, Detroit, Mich, USA, May 1999.

[23] R. S. Jamisola Jr., A. A. Maciejewski, and R. G. Roberts, "Failure-tolerant path planning for kinematically redundant manipulators anticipating locked-joint failures," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 603–612, 2006.

[24] B. Xie, J. Zhao, and Y. Liu, "Fault tolerant motion planning of robotic manipulators based on a nested RRT algorithm," *Industrial Robot*, vol. 39, no. 1, pp. 40–46, 2012.

[25] H. Abdi, S. Nahavandi, Y. Frayman, and A. A. MacIejewski, "Optimal mapping of joint faults into healthy joint velocity space for fault-tolerant redundant manipulators," *Robotica*, vol. 30, no. 4, pp. 635–648, 2012.

[26] H. Abdi and S. Nahavandi, "Joint velocity redistribution for fault tolerant manipulators," in *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics (RAM '10)*, pp. 492–497, IEEE, Singapore, June 2010.

[27] Z. Jing and F. Cheng, "On the joint velocity jump during fault tolerant operations for manipulators with multiple degrees of redundancy," *Mechanism and Machine Theory*, vol. 44, no. 6, pp. 1201–1210, 2009.

[28] Q. Jia, T. Li, G. Chen, H. Sun, and J. Zhang, "Velocity jump reduction for manipulator with single joint failure," in *Proceedings of the International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI '14)*, pp. 1–6, Beijing, China, September 2014.

[29] Y. Wu, F. Sun, J. Zheng, and Q. Song, "A robust training algorithm of discrete-time MIMO RNN and application in fault tolerant control of robotic system," *Neural Computing and Applications*, vol. 19, no. 7, pp. 1013–1027, 2010.

[30] Z. Li, "Application of fault tolerant controller based on RBF neural networks for mobile robot," in *Proceedings of the International Symposium on Intelligent Ubiquitous Computing and Education (IUCE '09)*, pp. 531–534, IEEE, Chengdu, China, May 2009.

[31] A. Noore, "Real time fault tolerant control of robot manipulators," *Mathematical and Computer Modelling*, vol. 38, no. 1-2, pp. 13–22, 2003.

[32] 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.

[33] H. Abdi, S. Nahavandi, and Y. Frayman, "A class of optimal fault tolerant Jacobian for minimal redundant manipulators based on symmetric geometries," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 1532–1537, IEEE, Anchorage, Alaska, USA, October 2011.