Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Failure-Tolerant Path Planning for the PA-10 Robot Operating Amongst Obstacles

Rodrigo S. Jamisola, Jr. & Anthony A. Maciejewski
Electrical & Computer Engineering Department
Colorado State University
Ft. Collins, Colorado 80523-1373, USA
E-mails: {r.jamisola, aam}@colostate.edu

Rodney G. Roberts
Electrical & Computer Engineering Department
Florida A&M-Florida State University
Tallahassee, Florida 32310-6046, USA
E-mail: rroberts@eng.fsu.edu

***Abstract*— This work considers kinematic failure tolerance when obstacles are present in the environment. An example is given using a fully spatial redundant robot, the seven degree-of-freedom Mitsubishi PA-10. This article addresses the issue of finding a collision-free path such that a redundant robot can successfully move from a start to a goal position and/or orientation in the workspace despite any single locked-joint failure at any time. An algorithm is presented that searches for a continuous obstacle-free monotonic surface in the configuration space that guarantees the existence of a solution. The method discussed is based on the following assumptions: a robot is redundant relative to its task, only a single locked-joint failure occurs at any given time, the robot is capable of detecting a joint failure and immediately locks the failed joint, and the environment is static and known.**

I. Introduction

Tasks carried out by robots in hazardous or remote environments preclude human intervention. A robot failure in a remote environment would mean an inability to immediately complete a desired task and an unexpected delay due to robot repair. In a hazardous environment, a robot failure during task execution could also pose significant danger. It is therefore helpful if a robot has the ability to gracefully recover from a failure and continue, albeit in a degraded manner, to complete the task at hand.

Failure-tolerant path planning is a robot motion planning strategy that gives redundant robots the ability to gracefully accommodate joint failures. This has been the focus of many studies in the past decade. Most of these studies have involved optimizing a robot configuration at a given time so that any joint failure would have the least impact on the robot operation. The earliest work on kinematic failure tolerance [1] used the minimum singular value of the manipulator Jacobian matrix as a worst-case measure of a robot's tolerance to a joint failure. The nature of robot joint failures that have been studied include locked-joint [2]–[4] and free-swinging joint failures [5]. A real-time implementation of kinematic failure tolerance was demonstrated in [4].

Other studies related to enhancing a robot's tolerance to failure include: failure detection [6]–[8], low-level failure avoidance and recoverability [9], [10], layered failure tolerance control structure [11]–[13], failure tolerance by trajectory planning [14], [15], and kinematic failure recovery [16].

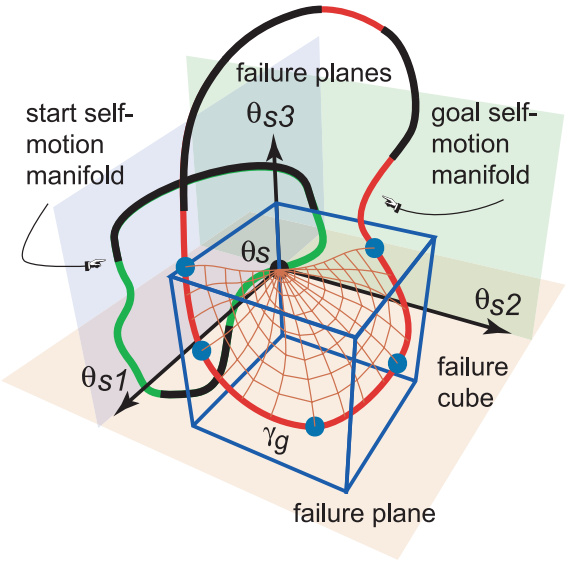

Fig. 1. The configuration space for a single degree of redundancy robot shown with a start and a goal self-motion manifold. All the failure planes corresponding to an obstacle-free start configuration, $\boldsymbol{\theta}_s$, intersect a continuous obstacle-free portion of the goal self-motion manifold, $\boldsymbol{\gamma}_g$. The failure cube contains $\boldsymbol{\theta}_s$ and $\boldsymbol{\gamma}_g$. The failure surface corresponding to $\boldsymbol{\theta}_s$, shown as a web-like network of paths, is identified by connecting $\boldsymbol{\theta}_s$ to points on $\boldsymbol{\gamma}_g$ via monotonic paths within the failure cube. Each node along $\boldsymbol{\gamma}_g$ defines an intersection with either a failure plane or a face of the failure cube.

Although the presence of obstacles in the environment greatly affects a kinematic failure tolerance algorithm, this issue has not been given much attention in the past. One of the earliest works in this area [14] exhaustively checked every possibility of failure at every instance in time as the robot plans to move from a start to a goal workspace location in order to guarantee collision-free paths for any joint failure that may occur. However, the proposed method is prohibitively expensive in terms of computation time. In more recent work [17], a method was introduced that searches for a continuous obstacle-free monotonic surface from the start to the goal in the configuration space. The existence of this obstacle-free surface guarantees that the given robot can successfully reach a goal workspace position and/or orientation from an obstacle-free start configuration despite any single locked-joint failure

at any time, without exhaustively checking every possibility of failure at every instance in time. In this work, the concept presented in [17] is expanded from a planar three degree-of-freedom robot to a seven degree-of-freedom redundant robot, the Mitsubishi PA-10.

This work proceeds as follows. Section II defines some important terms used in this paper. Section III states the conditions that guarantee the existence of a solution and the algorithm for finding a solution. Section IV presents methods for generating monotonic paths to identify a monotonic failure surface. Section V presents a global failure tolerance measure. Section VI presents the Mitsubishi PA-10 robot example. Lastly, Section VII gives the summary and conclusion of this work.

II. Definition of Terms

Let n denote the number of degrees of freedom (DOFs) of a robot and let m denote the number of DOFs of a robot's workspace. A robot is said to be kinematically redundant when $n > m$, and its degree of redundancy is $r = n - m$. For a kinematically redundant robot, a given end-effector position and/or orientation, denoted $\mathbf{x}$, generally corresponds to an infinite number of configurations in the configuration space (C-space). The set of configurations in C-space that result in the same $\mathbf{x}$ is called the pre-image of $\mathbf{x}$, denoted $f^{-1}(\mathbf{x})$. The pre-image can be written as a union of disjoint connected sets

$$f^{-1}(\mathbf{x}) = \bigcup_{i=1}^{n_m} \boldsymbol{\mathcal{M}}_i. \tag{1}$$

The symbol $\boldsymbol{\mathcal{M}}_i$ denotes the i-th r-dimensional self-motion manifold in the inverse kinematic pre-image such that $\boldsymbol{\mathcal{M}}_i \cap \boldsymbol{\mathcal{M}}_j = \emptyset$ when $i \neq j$, while n_m denotes the number of self-motion manifolds [18].

A start self-motion manifold, denoted $\boldsymbol{\mathcal{M}}_s$, corresponds to a start position and/or orientation, denoted $\mathbf{x}_s$, while a goal self-motion manifold, denoted $\boldsymbol{\mathcal{M}}_g$, corresponds to a goal position and/or orientation, denoted $\mathbf{x}_g$. Fig. 1 shows a pair of single dimensional start and goal self-motion manifolds for a robot with $r = 1$. The dark portions of the self-motion manifolds denote configurations of the robot that are in contact with obstacles. A continuous obstacle-free portion of the goal self-motion manifold is denoted $\boldsymbol{\gamma}_g$ while an obstacle-free start configuration is denoted $\boldsymbol{\theta}_s$. A point on $\boldsymbol{\gamma}_g$ is denoted $\boldsymbol{\theta}_g$.

A *failure hyperplane*, denoted $\mathbf{H}_i$, is an $(n-1)$-dimensional hyperplane resulting from a locked-joint failure of joint i. A failure hyperplane associated with $\boldsymbol{\theta}_s$, denoted $\mathbf{H}_i(\boldsymbol{\theta}_s)$, is given by

$$\mathbf{H}_i(\boldsymbol{\theta}_s) = \{\boldsymbol{\theta} \mid \theta_i = \theta_{si}\} \tag{2}$$

where θ_i denotes the i-th component of $\boldsymbol{\theta}$ in a failure-induced C-space and θ_{si} denotes the i-th component of $\boldsymbol{\theta}_s$. Fig. 1 shows $\boldsymbol{\theta}_s$ with its corresponding failure planes.

A *failure hypercube*, denoted $\mathbf{V}$, is a hypervolume in C-space that contains a $\boldsymbol{\theta}_s$ and a $\boldsymbol{\gamma}_g$ such that the failure hyperplane $\mathbf{H}_i(\boldsymbol{\theta}_s)$ intersects $\boldsymbol{\gamma}_g$ for $i = 1, \ldots, n$. A failure hypercube associated with $\boldsymbol{\theta}_s$ and $\boldsymbol{\gamma}_g$ has the form

$$\mathbf{V} = \{\boldsymbol{\theta} \mid \theta_{\mathbf{H}_i^l} \leq \theta_i \leq \theta_{\mathbf{H}_i^u} \text{ for } i = 1, \ldots, n\}. \tag{3}$$

Thus, a failure hypercube is composed of all configurations whose joint component θ_i lies within the bounds of the lower bounding failure hyperplane, denoted $\mathbf{H}_i^l$, and upper bounding failure hyperplane, denoted $\mathbf{H}_i^u$, for $i = 1, \ldots, n$ whose locations are defined by the values of $\theta_{\mathbf{H}_i^l}$ and $\theta_{\mathbf{H}_i^u}$, respectively. Two bounding failure hyperplanes are defined by the failure hyperplanes associated with $\boldsymbol{\theta}_s$ that first intersects $\boldsymbol{\gamma}_g$, denoted $\mathbf{H}_i^0(\boldsymbol{\theta}_s)$, and that last intersects $\boldsymbol{\gamma}_g$, denoted $\mathbf{H}_j^1(\boldsymbol{\theta}_s)$. All the other failure hyperplanes intersect $\boldsymbol{\gamma}_g$ within the bounds of these two bounding failure hyperplanes. The rest of the bounding failure hyperplanes are defined by the extremal points of $\boldsymbol{\gamma}_g$, namely, the lower bounding failure hyperplane,

$$\mathbf{H}_i^l = \{\boldsymbol{\theta} \mid \theta_{\mathbf{H}_i^l} = \min(\gamma_{gi})\}, \tag{4}$$

and the upper bounding failure hyperplane,

$$\mathbf{H}_i^u = \{\boldsymbol{\theta} \mid \theta_{\mathbf{H}_i^u} = \max(\gamma_{gi})\}. \tag{5}$$

Note that the entire volume of a failure hypercube is not typically obstacle free.

A *failure surface*, denoted $\boldsymbol{\mathcal{S}}$, is a continuous obstacle-free monotonic surface within a failure hypercube, $\mathbf{V}$, that contains $\boldsymbol{\theta}_s$ and is bounded by three curves: $\boldsymbol{\gamma}_g$, an obstacle-free monotonic curve lying in $\mathbf{H}_i^0(\boldsymbol{\theta}_s)$, and an obstacle-free monotonic curve lying in $\mathbf{H}_j^1(\boldsymbol{\theta}_s)$. Fig. 1 shows a web-like network of paths that represent a failure surface within a failure cube. It is identified by connecting monotonic paths from $\boldsymbol{\theta}_s$ to points on $\boldsymbol{\gamma}_g$. Straight-line connections are used to check for continuity of obstacle-free space between paths. The intersection of a monotonic surface with any hyperplane parallel to its failure hyperplane is a non-closed curve. Thus a monotonic surface, as shown in Fig. 2, does not have any local internal minima or maxima with regard to any θ_i-axis.

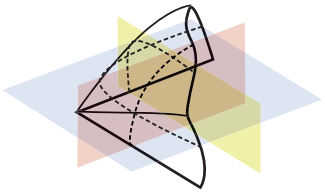

Fig. 2. A monotonic surface is defined as a surface where there are no closed contours along the intersection with any plane parallel to its failure planes. A monotonic surface has no internal local minima or maxima with regard to any θ_i-axis.

III. Conditions for the Existence of a Solution and Algorithm

In this section, two conditions are presented that would guarantee the existence of a solution to kinematic failure tolerance with obstacle avoidance. The algorithm of the proposed method is also presented.

A. Conditions

A necessary condition is derived to identify the set of feasible $\boldsymbol{\theta}_s$'s and eliminate those that are not feasible.

Necessary Condition. A given obstacle-free start configuration, $\boldsymbol{\theta}_s$, is considered a feasible start configuration if all the corresponding failure hyperplanes associated with $\boldsymbol{\theta}_s$ intersect a continuous obstacle-free portion of the goal self-motion manifold, $\boldsymbol{\gamma}_g$, that is,

$$\mathbf{H}_i(\boldsymbol{\theta}_s) \cap \boldsymbol{\gamma}_g \neq \emptyset, \text{ for all } i = 1, \ldots, n. \tag{6}$$

This ensures a possibility of reaching $\boldsymbol{\gamma}_g$ despite a joint failure at $\boldsymbol{\theta}_s$. Note that this is equivalent to $\mathbf{x}_g$ being in the fault-tolerant workspace [3] of $\mathbf{x}_s$. From all the feasible configurations, a sufficient condition is derived that guarantees the existence of a solution.

Sufficient Condition. Consider a given failure hypercube, $\mathbf{V}$, containing a feasible obstacle-free start configuration, $\boldsymbol{\theta}_s$, and a continuous obstacle-free portion of the goal self-motion manifold, $\boldsymbol{\gamma}_g$. If a failure surface, $\boldsymbol{\mathcal{S}}$, which is a continuous obstacle-free monotonic surface inside the failure hypercube, $\mathbf{V}$, exists, then an obstacle-free path to the goal is guaranteed for any single locked-joint failure at any given time despite the presence of obstacles in the workspace.

B. Algorithm

The step-by-step procedure for implementing failure-tolerant path planning with obstacle avoidance is enumerated in the following.

1. Determine $\boldsymbol{\mathcal{M}}_s$ and $\boldsymbol{\mathcal{M}}_g$ from the given $\mathbf{x}_s$ and $\mathbf{x}_g$, respectively.
2. Identify $\boldsymbol{\theta}_s$ and $\boldsymbol{\gamma}_g$.
3. Check for intersections of the failure hyperplane $\mathbf{H}_i(\boldsymbol{\theta}_s)$ with $\boldsymbol{\gamma}_g$ for $i = 1, \ldots, n$. (Note that this step uses the necessary condition in Section III-A).
4. Check for the existence of a failure surface, $\boldsymbol{\mathcal{S}}$. This is done by generating monotonic paths from feasible $\boldsymbol{\theta}_s$ to points on $\boldsymbol{\gamma}_g$. Straight-line connections between paths are used to check for the continuity of obstacle-free space between paths. The resulting obstacle-free web of paths represents the failure surface, $\boldsymbol{\mathcal{S}}$. (This step utilizes the sufficient condition in Section III-A).

Techniques presented in [19]–[21] could be used to check for collision-free space in monotonic path generation. The specific technique used in this work for monotonic path generation is discussed in the following section. Given that $\boldsymbol{\mathcal{S}}$ exists, to move from $\mathbf{x}_s$ towards $\mathbf{x}_g$, the manipulator configuration traverses from $\boldsymbol{\theta}_s$ along the continuous web of paths that represents $\boldsymbol{\mathcal{S}}$ toward $\boldsymbol{\gamma}_g$. Because $\boldsymbol{\mathcal{S}}$ is known to be collision free, as long as the manipulator configuration remains on the surface, the manipulator would be free from collision and at the same time it can reach the goal for any single locked-joint failure at any time. If no $\boldsymbol{\mathcal{S}}$ exists, then it is not guaranteed that the robot can successfully complete its task for any single locked-joint failure with the given obstacles in the environment.

The computational complexity of the proposed algorithm is highly dependent on the method used for computing the start and goal self-motion manifolds, and the method used for collision detection. For $r = 1$, the computational complexity is $O(mn^2) + O(mnp)$ where p is the number of obstacles in the workspace. The first term is the contribution for the computation of the self-motion manifolds, while the second term is the contribution due to collision detection.

IV. Generating Monotonic Paths

Parametric monotonic polynomials $p(t)$ are used to generate paths from $\boldsymbol{\theta}_s$ to $\boldsymbol{\gamma}_g$. Three types of parametric monotonic polynomials are used: linear, quadratic, and cubic.

A polynomial $p(t)$ is monotonic on a region of interest provided that its derivative, $p'(t)$, does not change sign in that region. In particular, a polynomial $p(t)$ is monotonic on $[0, 1]$ provided that the zeros of $p'(t)$ do not occur in the open interval $(0, 1)$. A linear parametric equation $p(t) = at + b$ is monotonic throughout because its derivative $p'(t) = b$ is constant.

A. Generating Monotonic Quadratic Polynomials

It is easy to see that any quadratic polynomial $p(t)$ satisfying the constraints $p(0) = \theta_{si}$ and $p(1) = \theta_{gi}$ has the form

$$p(t) = \theta_{si}(1 - t) + \theta_{gi}t + \alpha t(1 - t) \tag{7}$$

where the parameter α can be any real number and θ_{gi} denotes the i-th component of $\boldsymbol{\theta}_g$. Now a polynomial is monotonic on the closed interval $[0, 1]$ if and only if its derivative does not change sign on the open interval $(0, 1)$. Since the derivative of a quadratic polynomial represents the equation of a line, it follows that we only need to check the endpoints of the interval $[0, 1]$ to determine the set of α's that makes (7) monotonic on $[0, 1]$. Therefore, for (7) to be monotonic on $[0, 1]$, $p'(0) = \theta_{gi} - \theta_{si} + \alpha$ and $p'(1) = \theta_{gi} - \theta_{si} - \alpha$ should not have opposite signs. This is clearly true if and only if $|\alpha| \leq |\theta_{gi} - \theta_{si}|$.

B. Generating Monotonic Cubic Polynomials

There are two cases characterizing when $p(t)$ is a cubic polynomial that is monotonic on $[0, 1]$ with end conditions $p(0) = \theta_{si}$ and $p(1) = \theta_{gi}$.

Case 1. The roots of $p'(t)$ are complex or occur as a double root. This occurs if and only if $p(t)$ is monotonic everywhere. In this case $p'(t) = K(t^2 + 2at + a^2 + b^2)$ where a is any real number and $b \geq 0$. We can parameterize a and b by $a = \tan\phi_1$ for $-\pi/2 < \phi_1 < \pi/2$ and $b = \tan\phi_2$ for $0 \leq \phi_2 < \pi/2$. Integrating $p'(t)$ and taking into account the endpoints we obtain

$$p(t) = (\theta_{gi} - \theta_{si})\frac{2t^3 + 3at^2 + 6(a^2 + b^2)t}{2 + 3a + 6(a^2 + b^2)} + \theta_{si}. \tag{8}$$

Case 2. The roots of $p'(t)$ are real and do not occur in $(0, 1)$. Then $p'(t) = K(t - t_1)(t - t_2)$ where $t_i \notin (0, 1)$, $i = 1, 2$. There are three subcases: (2a) $t_1, t_2 \in (-\infty, 0]$; (2b) $t_1 \in (-\infty, 0]$ and $t_2 \in [1, \infty)$; and (2c) $t_1, t_2 \in [1, \infty)$. These

cases can be parameterized in a manner similar to that used in Case 1. Integrating $p'(t)$ and satisfying the endpoint conditions gives

$$p(t) = (\theta_{gi} - \theta_{si}) \frac{2t^3 - 3(t_1 + t_2)t^2 + 6t_1t_2t}{2 - 3(t_1 + t_2) + 6t_1t_2} + \theta_{si}. \quad (9)$$

V. A Global Kinematic Failure Tolerance Measure

In this section, we will present a global kinematic failure tolerance measure which is derived from the self-motion manifold of a manipulator. Based on this measure, an estimate can be deduced as to how tolerant a given manipulator would be to any joint failure throughout its workspace. It can also determine if a given manipulator is always intolerant to a specific joint failure for any configuration.

A self-motion manifold, $\boldsymbol{\mathcal{M}}$, can be derived by integrating the null space component, $\mathcal{N}(\mathbf{J})$, of the manipulator Jacobian matrix, $\mathbf{J}(\boldsymbol{\theta})$. The singular value decomposition (SVD) of $\mathbf{J}(\boldsymbol{\theta})$ can be used to compute the null space matrix, $\mathbf{N}_J$, which represents $\mathcal{N}(\mathbf{J})$. Generically, the dimension of $\mathbf{N}_J$ is equal to r. Each column of $\mathbf{N}_J$, defined as $\hat{\mathbf{n}}_J^k$ where $k = 1, \ldots, r$, forms an orthonormal basis of $\mathcal{N}(\mathbf{J})$. For $r = 1$, $\mathbf{N}_J$ is a vector and can be expressed as $\hat{\mathbf{n}}_J$.

It has been shown in [2] that, for $r = 1$, a configuration $\boldsymbol{\theta}$ is intolerant to a joint i failure if the null vector component $\hat{n}_{Ji} = 0$. If this is true for all configurations, then the robot is globally intolerant to an i-th joint failure. A joint's tolerance to failure is related to its range of values in the self-motion manifold. The larger the change in a joint's value, the greater tolerance there will be to a failure in that joint. A joint whose value is constant throughout the self-motion manifold is a globally failure intolerant joint and is a critical manipulator joint.

A global kinematic failure tolerance measure τ can be derived by using the range of each joint's excursion in the self-motion manifold. The joint with the minimum range of excursion determines the worst-case global failure tolerance measure

$$\tau_{min} = \min_{1 \le i \le n} (|\Delta\theta_i|) \quad (10)$$

where $\Delta\theta_i$ is the maximum excursion of joint i as it spans the self-motion manifold. (This is the same as the bounding box discussed in [3].) Other possible global failure tolerance measures include the square root of the sum of squares of each joint's maximum excursion in the self-motion manifold

$$\tau_{ave} = \sqrt{\sum_{i=1}^{n} \Delta\theta_i^2} \quad (11)$$

or the ratio of the joint with the least maximum excursion with that of the joint with the largest maximum excursion in the self-motion manifold

$$\tau_{ratio} = \frac{\min_{1 \le i \le n}(|\Delta\theta_i|)}{\max_{1 \le i \le n}(|\Delta\theta_i|)} \quad (12)$$

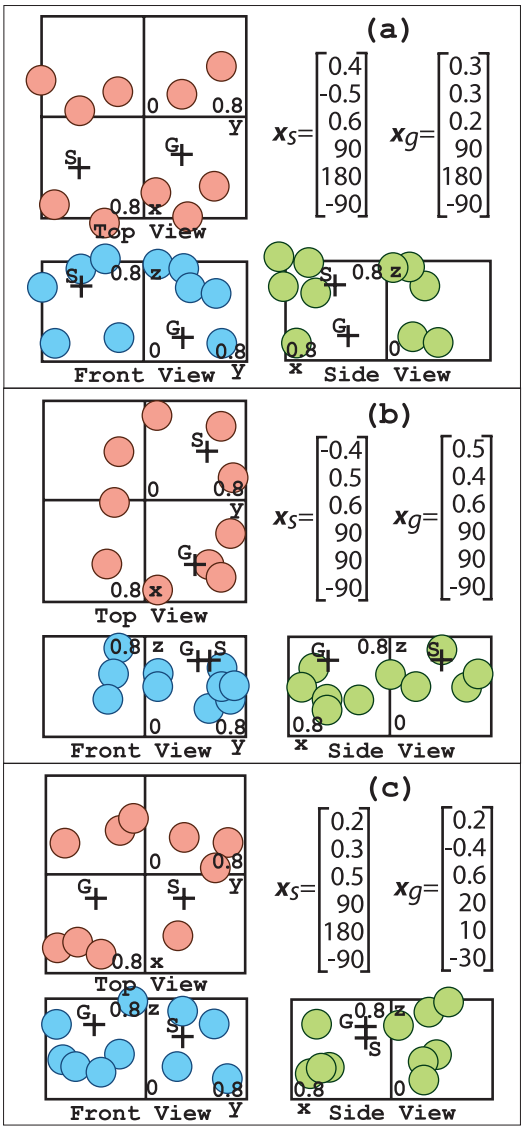

Fig. 3. The Mitsubishi PA-10 workspace with 10 randomly generated spherical obstacles of diameter 0.254 m (10 in) for the corresponding $\mathbf{x}_s$ and $\mathbf{x}_g$ shown. The corresponding solutions are shown in Table I. The first three vector components are the desired positions in units of meters. The last three vector components are the desired orientations in units of degrees expressed in terms of Euler angles using System I as discussed in [22].

or the product of each joint's maximum excursion in the self-motion manifold

$$\tau_{prod} = \prod_{i=1}^{n} |\Delta\theta_i|. \quad (13)$$

Normally, this range can vary from zero to some finite non-negative value. It would be computationally expensive to find the exact value of τ for a given robot because one would need to evaluate the whole space of the self-motion manifold. However, if $\tau_{min} = 0$, then $\Delta\theta_i = 0$ for some i and the robot is globally intolerant to a failure of joint i, i.e., a critical manipulator joint. A good redundant manipulator design that considers failure tolerance is a design with a large global

TABLE I

THE SET OF SOLUTIONS FOR THE EXAMPLES SHOWN IN FIG. 3 WITH THEIR CORRESPONDING FEASIBLE OBSTACLE-FREE START CONFIGURATION, $\boldsymbol{\theta}_s$. THE FAILURE HYPERPLANE $\mathbf{H}_i(\boldsymbol{\theta}_s)$ INTERSECTS $\boldsymbol{\gamma}_g$ AT PARAMETERIZATION INTERVAL $[0, 1]$.

Fig. 3	Feasible Obstacle-Free Start Configuration, $\boldsymbol{\theta}_s$							Failure Hyperplane $\mathbf{H}_i(\boldsymbol{\theta}_s)$ Intersecting $\boldsymbol{\gamma}_g$						
	$\boldsymbol{\theta}_{s1}$	$\boldsymbol{\theta}_{s2}$	$\boldsymbol{\theta}_{s3}$	$\boldsymbol{\theta}_{s4}$	$\boldsymbol{\theta}_{s5}$	$\boldsymbol{\theta}_{s6}$	$\boldsymbol{\theta}_{s7}$	$\mathbf{H}_1$	$\mathbf{H}_2$	$\mathbf{H}_3$	$\mathbf{H}_4$	$\mathbf{H}_5$	$\mathbf{H}_6$	$\mathbf{H}_7$
(a)	-11.5	75.1	43.8	-49.2	46.8	-117.9	27.4	1.0	0.6	0.2	n.a.	0.4	0.8	0.0
(b)	-6.4	-76.6	50.7	52.1	33.0	110.1	28.7	0.0	0.8	0.2	n.a.	0.4	0.6	1.0
(c)	-0.5	17.5	235.7	-87.6	15.1	-128.1	191.1	0.6	0.8	1.0	n.a.	0.4	0.2	0.0

failure tolerance measure τ. Such a design would allow a larger range of values in the manipulator's self-motion manifold and increases the likelihood of satisfying the necessary condition in Section III-A.

There is a rare exception that a robot can still be failure tolerant to a joint i failure when $\tau_{min} = 0$ due to $\Delta\theta_i = 0$. This happens when the projections of the start and goal self-motion manifolds along the critical manipulator joint i axis coincide with each other. This means that for the robot to move from $\mathbf{x}_s$ to $\mathbf{x}_g$, joint i needs to be at this fixed angle where the manifolds' projections coincide, thus, a locked-joint failure at joint i would still make the robot tolerant to a joint i failure.

VI. PA-10 SEVEN DOF REDUNDANT ROBOT EXAMPLE

The Mitsubishi PA-10 robot is not fully a kinematically failure-tolerant robot because the null space component of joint four is zero, i.e.,

$$\hat{n}_{J4} = 0, \tag{14}$$

throughout the entire C-space. Thus, the PA-10 has a worst-case global failure tolerance measure $\tau_{min} = 0$ due to joint four and is intolerant to a joint four failure. A good failure-tolerant robot design includes link offsets which help in creating excursion in the manipulator's self-motion manifold. For the PA-10, the axes of the shoulder, i.e., joints one, two, and three meet at a common point as do the axes of the wrist, i.e., joints five, six, and seven. Because joint four is the only joint that can alter the distance between these two common points, it is a critical manipulator joint. Hence, the PA-10 is intolerant to a joint four failure. However, we chose to present the PA-10 as an example because it is the most common commercially available fully spatial redundant robot on the market. In this section, we will show that if joint four does not fail, the PA-10 is failure tolerant to a single locked-joint failure of any of the remaining six joints. In the following examples, the simulation program for the PA-10 does not consider the joint limits and self collision. The end-effector offset from the wrist link is set at 0.3 m.

Ten spherical obstacles with a diameter of 0.254 m (10 in) are randomly placed in the PA-10 workspace with a given set of $\mathbf{x}_s$ and $\mathbf{x}_g$. The equivalent $\boldsymbol{\mathcal{M}}_s$ and $\boldsymbol{\mathcal{M}}_g$ are then computed and the corresponding $\boldsymbol{\theta}_s$ and $\boldsymbol{\gamma}_g$ are respectively determined. A candidate $(\boldsymbol{\theta}_s, \boldsymbol{\gamma}_g)$ pair that satisfies the necessary condition is identified when the failure hyperplane $\mathbf{H}_i(\boldsymbol{\theta}_s)$ intersects

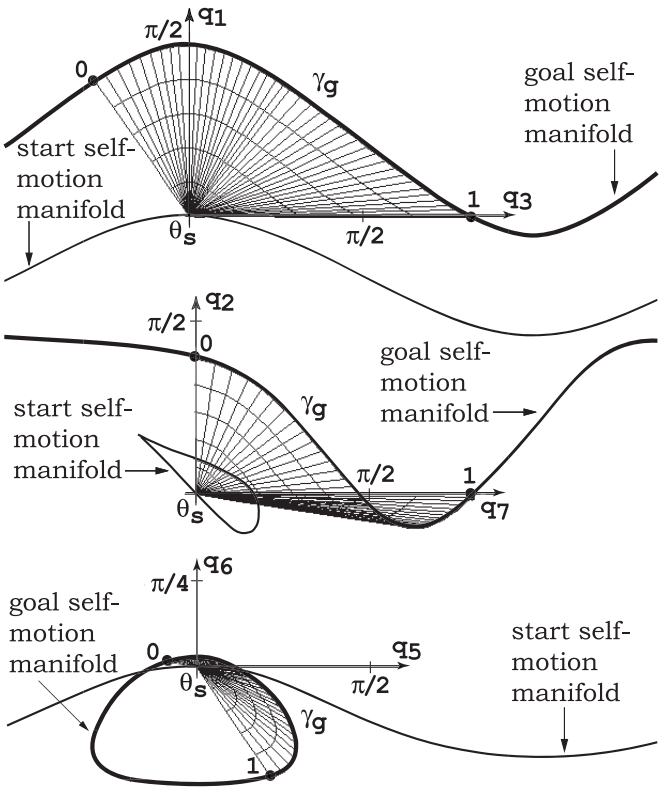

Fig. 4. The failure surface for example (a), shown as a web of paths in the configuration space, with projections from joint axes 1 and 3, 2 and 7, and 5 and 6. The projections are shown in the same scale with units of radians. The bold curves represent portions of $\boldsymbol{\mathcal{M}}_g$, while the less thick curves represent portions of $\boldsymbol{\mathcal{M}}_s$. The axes shown are translated from the origin to the feasible $\boldsymbol{\theta}_s$. Its corresponding $\boldsymbol{\gamma}_g$ is the curve between the points labeled "0" and "1".

$\boldsymbol{\gamma}_g$ for $i = 1, \ldots, n$. Using this $(\boldsymbol{\theta}_s, \boldsymbol{\gamma}_g)$ pair, the algorithm searches for the existence of a failure surface, $\boldsymbol{\mathcal{S}}$. This is done by connecting $\boldsymbol{\theta}_s$ to points on $\boldsymbol{\gamma}_g$. For each point on $\boldsymbol{\gamma}_g$, the algorithm first attempts to use a linear path. If the linear path is not obstacle free, it tries to use a monotonic quadratic path using the possible values of α in (7) until an obstacle-free monotonic quadratic path is found. When all the possible values of α are tried and no such path can be found, the algorithm discards this $(\boldsymbol{\theta}_s, \boldsymbol{\gamma}_g)$ pair and uses the next $(\boldsymbol{\theta}_s, \boldsymbol{\gamma}_g)$ pair that satisfies the necessary condition. If all such pairs are exhausted without completing a failure surface, $\boldsymbol{\mathcal{S}}$, then the algorithm terminates with a message that is was unsuccessful.

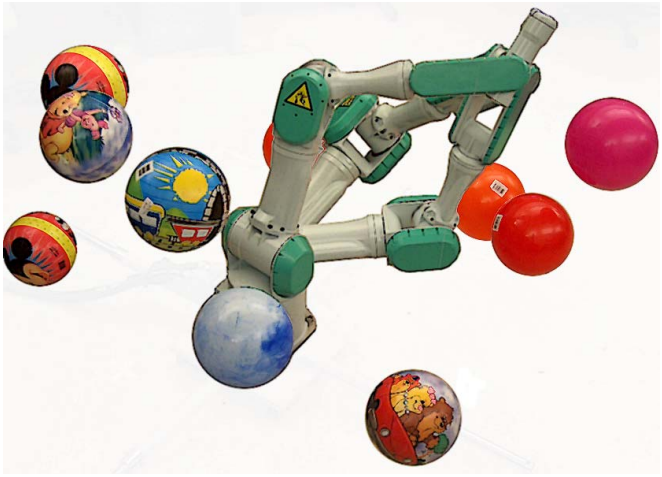

Fig. 5. Snapshots of the Mitsubishi PA-10 robot amongst obstacles while going through a portion of γ_g for example (c).

To check for continuity of free space between connected paths, straight-line connections between paths are used. If the path connections from $\boldsymbol{\theta}_s$ to all points on γ_g parameterized between [0,1] and the straight-line connections between paths are obstacle free, a failure surface, $\boldsymbol{\mathcal{S}}$, is successfully identified.

Three examples are shown in Fig. 3 where the corresponding failure surfaces, $\boldsymbol{\mathcal{S}}$, are successfully found. The first example shows the PA-10 with a start and goal end-effector orientations facing downward with the corresponding desired positions. The second example shows a start and goal end-effector orientations facing forward, while the third example shows a start end-effector orientation facing downward and an arbitrary final end-effector orientation for the corresponding desired positions. The set of solutions are shown in Table I with the corresponding feasible $\boldsymbol{\theta}_s$. The failure hyperplane $\mathbf{H}_i(\boldsymbol{\theta}_s)$ intersects γ_g at the shown values of γ_g parameterized on the interval $[0, 1]$. Recall that the failure hyperplane corresponding to joint four, $\mathbf{H}_4(\boldsymbol{\theta}_s)$, does not intersect γ_g which thus makes joint four a critical manipulator joint. Fig. 4 shows the corresponding failure surface, $\boldsymbol{\mathcal{S}}$, for example (a). Fig. 5 shows snapshots of the PA-10 amongst obstacles while going through a portion of γ_g in example (c).

VII. Summary and Conclusion

This work considers the problem of guaranteeing failure tolerance when obstacles are present in the environment. It has been shown that it is possible to guarantee that a robot can successfully reach the goal workspace position and/or orientation, $\mathbf{x}_g$, despite any single locked-joint failure without exhaustively checking every possibility of failure at every instance in time. Conditions were formulated that guarantee the existence of a solution. An algorithm was presented that searches for a continuous obstacle-free monotonic surface in the configuration space, called a failure surface, $\boldsymbol{\mathcal{S}}$, whose existence guarantees the solution. A global failure tolerance measure was also defined that can determine if a robot is globally failure intolerant to a particular joint failure.

References

[1] A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, Cincinnati, OH, May 13-18 1990, pp. 638–642.

[2] R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 543–552, Aug. 1996.

[3] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robot. Automat.*, vol. 13, no. 4, pp. 622–629, Aug. 1997.

[4] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure-tolerant control of kinematically redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 15, no. 6, pp. 1109–1116, Dec. 1999.

[5] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Trans. Robot. Automat.*, vol. 14, no. 4, pp. 566–575, Aug. 1998.

[6] M. L. Visinsky, I. D. Walker, and J. R. Cavallaro, "New dynamic model-based fault detection threshold for robot manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Diego, CA, May 8-13 1994, pp. 1388–1395.

[7] L. S. Lopes and L. M. Camarinha-Matos, "A machine learning approach to error detection and recovery in assembly," in *Proc. Int. Conf. Intell. Robots Syst.*, Pittsburgh, PA, Aug. 5-9 1995, pp. 197–203.

[8] H. Schneider and P. M. Frank, "Fuzzy logic based threshold adaption for fault detection in robots," in *Proc. 3rd IEEE Conf. Contr. Applicat.*, Glasgow, UK, Aug. 24-26 1994, pp. 1127–1132.

[9] T. S. Wikman, M. S. Branicky, and W. S. Newman, "Reflexive collision avoidance: A generalized approach," in *Proc. IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 2-6 1993, pp. 31–36.

[10] Y. Ting, S. Tosunoglu, and R. Freeman, "Actuator saturation avoidance for fault-tolerant robots," in *Proc. 32nd Conf. Decision Contr.*, San Antonio, TX, Dec. 1993, pp. 2125–2130.

[11] Y. Ting, S. Tosunoglu, and D. Tesar, "A control structure for fault-tolerant operation of robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 2-6 1993, pp. 684–690.

[12] K. S. Tso, M. Hecht, and N. I. Marzwell, "Fault-tolerant robotic system for critical applications," in *Proc. IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 2-6 1993, pp. 691–696.

[13] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Automat.*, vol. 11, no. 4, pp. 477–490, Aug. 1995.

[14] C. J. J. Paredis and P. K. Khosla, "Fault tolerant task execution through global trajectory planning," *Rel. Eng. Syst. Safety*, vol. 53, pp. 225–235, 1996.

[15] S. K. Ralph and D. K. Pai, "Computing fault tolerant motions for a robot manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, Detroit, MI, May 10-15 1999, pp. 486–493.

[16] J. Park, W.-K. Chung, and Y. Youm, "Failure recovery by exploiting kinematic redundancy," in *5th Int. Workshop Robot Human Commun.*, Tsukuba, Japan, Nov. 11-14 1996, pp. 298–305.

[17] R. S. Jamisola, Jr., A. A. Maciejewski, and R. G. Roberts, "A path planning strategy for kinematically redundant manipulators anticipating joint failures in the presence of obstacles," in *Proc. Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, Oct. 27-31 2003, pp. 142–148.

[18] J. W. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds," in *Proc. IEEE Int. Conf. Robot. Automat.*, Scottsdale, AZ, May 14-19 1989, pp. 264–270.

[19] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, Fall 1985.

[20] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[21] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[22] K. S. Fu, R. C. Gonzales, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. U.S.A.: McGraw-Hill, Inc., 1987.