Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

A Path Planning Strategy for Kinematically Redundant Manipulators Anticipating Joint Failures in the Presence of Obstacles

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

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

***Abstract*— This work considers the failure tolerant operation of a kinematically redundant manipulator in an environment containing obstacles. In particular, the article addresses the problem of planning a collision-free path for a manipulator operating in a static environment such that the manipulator can reach its desired goal despite a single locked-joint failure and the presence of obstacles in the environment. A method is presented that searches for a continuous obstacle-free space between the starting configuration and the desired final end-effector position which is characterized in the joint space by the goal self-motion manifold. This method guarantees completion of critical tasks in the event of a single locked-joint failure in the presence of obstacles.**[1]

I. Introduction

Robot reliability has been given increased attention throughout the last decade. This comes as a result of an increased demand for robots that can perform tasks in hazardous or remote environments that preclude human intervention. Examples of such tasks include space exploration [1], [2], underwater exploration [3], and nuclear waste disposal [4], [5], [6]. When a greater part of the work responsibility is passed onto the robot, the successful completion of a desired task greatly depends on the robot's reliability. However, robot failures are not uncommon. It has been estimated that 28.7% of industrial robots had a mean-time-between-failure of 100 hours or less, and the Japanese Ministry of Labor reported that over 60% of industrial robots had a mean-time-between-failure of less than 500 hours [7].

A number of articles have focused on robot reliability assessment and analysis [8], [9]. A number of methods have been proposed for analyzing and improving robot reliability including the use of fault trees [10], [11], [12], neural networks [13], kinematic redundancy [6], [14], [15], [16], and empirical formulas [17], [18], [19]. Robot designs for reliability are described in [4], [20], [21], while robot reliability experiments are performed in [2], [22].

While one would obviously want to design a manipulator to be as reliable as possible, failures will inevitably occur. It is therefore important to develop control strategies that will allow the robot to continue its task, albeit in a degraded manner, when a failure does occur. One approach to this is kinematic failure tolerance control. This type of control scheme configures the robot in anticipation of joint failures so that when a failure occurs, the robot can gracefully recover and continue on to complete its task [15]. A number of failure tolerant control strategies have been proposed, including adaptive control [23], [24], [25], reflex control [26], motion planning [27], [28], and least squares approaches [29]. Another approach is to determine a measure such as the kinematic failure tolerance measure (kfm) to quantify a robot's failure tolerance at a given configuration [30]. The failure tolerant control scheme would then try to optimize this value to maximize a robot's tolerance to a failure at a given manipulator configuration. Types of failure modes considered include both locked-joint [31], [16], [32] and free-swinging joint failures [33]. A real-time implementation of kinematic failure tolerant control is demonstrated in [31].

In this paper, we address the issue of kinematic failure tolerant control when obstacles are present in the environment. Although obstacle avoidance is a critical problem, most studies on kinematic failure tolerance have not considered the presence of obstacles in the environment. One of the earliest works that did consider obstacles in failure tolerant control is [27]. In [27], every possible configuration due to a joint failure was exhaustively checked in order to guarantee the existence of a collision-free path. The approach presented here is based on guaranteeing that such post-failure collision-free paths exist, without explicitly checking every possibility.

The paper proceeds as follows. Section II introduces the relevant definitions and the assumptions of the paper. Section III states a formal definition of the problem and the conditions for a solution. Section IV details the step-by-step process for a path planning strategy that guarantees failure tolerance, despite the presence of obstacles, to any single locked-joint failure. Section V illustrates an implementation of the proposed method on a three degree-

[1]Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

of-freedom planar robot with circular disk obstacles scattered in the environment. Lastly, Section VI contains the conclusions of this work.

II. Definition of Terms and Assumptions

A. Definitions and Terminology

We begin by defining the terminology and notation that is used in this article. These terms are related to the manipulator's workspace and joint space.

The joint space, also called the configuration space, characterizes the joint configuration of the robot. The configuration space (C-space) is formally defined as an n-dimensional space where n is the number of degrees of freedom (DOFs) of the robot. The components of a configuration in the C-space correspond to the respective values of the individual robot joints. In this way a point in the C-space defines a robot configuration. The mapping of points in the C-space to points in the workspace is called the kinematic function. The kinematic function $\mathbf{f}$ relates the joint configuration $\boldsymbol{\theta}$ to the end-effector position $\mathbf{x}$, i.e., $\mathbf{x} = \mathbf{f}(\boldsymbol{\theta})$. The quantity $\mathbf{x}$ is an m-dimensional vector where m is equal to the number of degrees of freedom in the workspace.

A manipulator is said to be kinematically redundant if $m < n$, in which case the degree of redundancy is equal to $r = n - m$. Because a kinematically redundant manipulator has, by definition, extra degrees of freedom, there are generally an infinite number of configurations corresponding to the same end-effector location. The set of configurations in C-space that result in the same end-effector location $\mathbf{x}$ is called the pre-image of $\mathbf{x}$. More specifically, the pre-image of $\mathbf{x}$ is the set $\mathbf{f}^{-1}(\mathbf{x}) = \{\boldsymbol{\theta} \mid \mathbf{x} = \mathbf{f}(\boldsymbol{\theta})\}$. The pre-image can be written as a union of disjoint connected sets

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

where $\boldsymbol{\mathcal{M}}_i$ is the i-th r-dimensional self-motion manifold in the inverse kinematic pre-image, $\boldsymbol{\mathcal{M}}_i \cap \boldsymbol{\mathcal{M}}_j = \emptyset$ when $i \neq j$, and n_{sm} is the number of self-motion manifolds [34]. A self-motion manifold corresponding to a start position $\mathbf{x}_s$ is called a start self-motion manifold $\boldsymbol{\mathcal{M}}_s$ while a self-motion manifold corresponding to a goal position $\mathbf{x}_g$ is called a goal self-motion manifold $\boldsymbol{\mathcal{M}}_g$.

The next definition relates to the structure of the reachable C-space following a locked-joint failure. When a locked-joint failure occurs, the failed joint can no longer be moved so the corresponding component in the C-space remains fixed. In other words, the reachable configurations following a joint failure are contained in an $(n-1)$-dimensional hyperplane in the C-space, called a *failure hyperplane*. For a single locked-joint failure at the configuration $\boldsymbol{\theta}_s$, a failure hyperplane $\mathbf{H}_i$ is given by

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

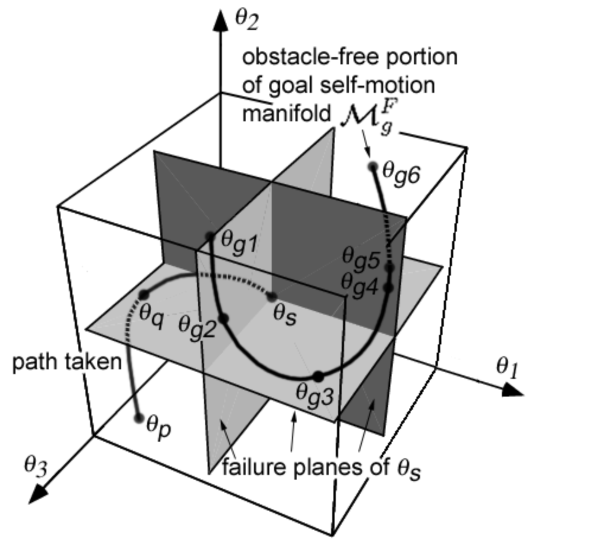


Fig. 1. A volume of obstacle-free C-space for a 3-DOF planar robot showing a portion of the obstacle-free goal self-motion manifold $\boldsymbol{\mathcal{M}}_g^F$ and obstacle-free configurations. All the failure planes of configuration $\boldsymbol{\theta}_s$ intersect $\boldsymbol{\mathcal{M}}_g^F$ which makes successfully reaching the goal self-motion manifold possible for any joint failure at $\boldsymbol{\theta}_s$. Configurations $\boldsymbol{\theta}_{g1}$, $\boldsymbol{\theta}_{g3}$, and $\boldsymbol{\theta}_{g5}$ are contained in the failure planes of $\boldsymbol{\theta}_s$. Configuration $\boldsymbol{\theta}_{g2}$ is contained in the bounding plane perpendicular to the $\boldsymbol{\theta}_3$-axis, $\boldsymbol{\theta}_{g4}$ is contained in the bounding plane perpendicular to the $\boldsymbol{\theta}_1$-axis, and $\boldsymbol{\theta}_{g6}$ is contained in the bounding plane perpendicular to the $\boldsymbol{\theta}_3$-axis.

where θ_i denotes the i-th component of $\boldsymbol{\theta}$ in a failure induced C-space, θ_{si} is the fixed value of the locked i-th component of configuration $\boldsymbol{\theta}_s$, and $i = 1, \ldots, n$. (See Fig. 1 for a 3-DOF C-space where configuration $\boldsymbol{\theta}_s$ is shown with its corresponding failure planes.)

A *failure hypercube* is a hypercube in C-space that includes an obstacle-free start configuration $\boldsymbol{\theta}_s$ and a portion of the obstacle-free goal self-motion manifold $\boldsymbol{\mathcal{M}}_g^F$ such that all the failure hyperplanes corresponding to the start configuration $\boldsymbol{\theta}_s$ intersect $\boldsymbol{\mathcal{M}}_g^F$. The F in the superscript denotes the fact that the manifold is obstacle-free. A failure hypercube has the form

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

where $\mathbf{V}_{sg}$ is the failure hypercube associated with the start configuration $\boldsymbol{\theta}_s$ and goal self-motion manifold $\boldsymbol{\mathcal{M}}_g$, $\mathbf{H}_i^l$ is the lower bounding hyperplane and $\mathbf{H}_i^u$ is the upper bounding hyperplane such that

$$\mathbf{H}_i^l = \{\boldsymbol{\theta} \mid \theta_i = \theta_{\mathbf{H}_i^l}\} \tag{4}$$

$$\mathbf{H}_i^u = \{\boldsymbol{\theta} \mid \theta_i = \theta_{\mathbf{H}_i^u}\} \tag{5}$$

for $i = 1, \ldots, n$. The values of $\theta_{\mathbf{H}_i^l}$ and $\theta_{\mathbf{H}_i^u}$ define the location of their corresponding hyperplanes. Note that a failure hypercube is not necessarily obstacle-free.

The bounding hyperplanes for a failure hypercube are defined as follows. Two bounding hyperplanes are defined by the two failure hyperplanes of $\boldsymbol{\theta}_s$ that contain the end points of $\mathcal{M}_g^F$. (See Fig. 1 where $\boldsymbol{\theta}_{g1}$ and $\boldsymbol{\theta}_{g5}$ are contained in the failure planes of $\boldsymbol{\theta}_s$.) The rest of the bounding hyperplanes are defined by the extremal points of $\mathcal{M}_g^F$. By an extremal point of $\mathcal{M}_g^F$, we mean a configuration with the property that one of its components is at a minimum or maximum. For example, $\boldsymbol{\theta}_{g3}$ is an extremal point of $\mathcal{M}_g^F$ as its second component is at a minimum. The i-th bounding hyperplane is tangent to $\mathcal{M}_g^F$ at an extremal point. (See Fig. 1 where the bounding plane perpendicular to the $\boldsymbol{\theta}_3$-axis is tangent to $\mathcal{M}_g^F$ at $\boldsymbol{\theta}_{g2}$, the failure plane is tangent to $\mathcal{M}_g^F$ at $\boldsymbol{\theta}_{g3}$, and the bounding plane perpendicular to the $\boldsymbol{\theta}_1$-axis is tangent to $\mathcal{M}_g^F$ at $\boldsymbol{\theta}_{g4}$.) An end-point of $\mathcal{M}_g^F$ can be an extremal point if it is a minimum or maximum point which in turn defines the lower or upper bounding hyperplane, respectively. (See Fig. 1 where $\boldsymbol{\theta}_{g1}$ is contained by the bounding plane perpendicular to the $\boldsymbol{\theta}_2$-axis.) Although the method is illustrated for a 3-DOF robot example for easy visualization, the method is fully general and can be applied to robots with higher DOFs.

B. Assumptions

Throughout the article we will make the following assumptions. First, the manipulators under consideration are redundant relative to the specified task. For example, a planar 3-DOF manipulator is redundant relative to positioning tasks in a planar environment while for fully spatial tasks, at least seven degrees of freedom are needed for the robot to be redundant. Although robot failures can occur at random with a possibility of multiple failures occurring at the same time, we will assume that only a *single* joint failure occurs at a given time. We further assume that the joint is locked at the instant of failure. It is assumed that the robot is capable of detecting a joint failure and the failed joint is immediately locked, either due to the failure itself or to the application of fail-safe brakes. The final assumption is that the environment is static and known.

III. Definition of the Problem and Conditions for the Existence of a Solution

In this section, we will formally define the problem addressed in the paper and state the conditions for a solution to the given problem. When joint failures are highly probable, an obstacle avoidance algorithm for path planning that does not consider failure tolerance will not be able to guarantee task completion. At the same time, a failure tolerance control algorithm that does not consider obstacles in the environment may also be inadequate. An example of a path planning algorithm that does not consider the possibility of a joint failure is discussed in the following to illustrate the problem created when a joint failure occurs.

Consider a start position $\mathbf{x}_s$, a goal position $\mathbf{x}_g$, and a collection of obstacles scattered in the robot workspace. Consider further an impending single joint failure. The problem is reformulated in C-space where the start position $\mathbf{x}_s$ is transformed to a start self-motion manifold $\mathcal{M}_s$ and the goal position $\mathbf{x}_g$ is transformed to a goal self-motion manifold $\mathcal{M}_g$. The obstacle avoidance algorithm then identifies an obstacle-free portion of C-space that includes a start configuration $\boldsymbol{\theta}_s$ and a part of the obstacle-free goal self-motion manifold $\mathcal{M}_g^F$. Fig. 1 illustrates such an algorithm for a 3-DOF planar robot where the shown C-space volume is obstacle-free. Obstacles are present beyond this given obstacle-free volume. The robot then starts from configuration $\boldsymbol{\theta}_p$ at time $t_p = 0$ and moves toward the goal self-motion manifold along the path shown. Configurations $\boldsymbol{\theta}_{g1}$ and $\boldsymbol{\theta}_{g6}$ are the end points of $\mathcal{M}_g^F$ contained inside the obstacle-free C-space volume. At time $t = t_s$, when the robot is at configuration $\boldsymbol{\theta}_s$, a joint failure occurs. Note that the three failure planes of configuration $\boldsymbol{\theta}_s$ intersect $\mathcal{M}_g^F$ at configurations $\boldsymbol{\theta}_{g1}$, $\boldsymbol{\theta}_{g3}$, and $\boldsymbol{\theta}_{g5}$, respectively. Thus for any joint failure at configuration $\boldsymbol{\theta}_s$, the failure induced C-space would be the corresponding failure plane at $\boldsymbol{\theta}_s$ which provides a possibility of reaching the goal because it contains a point of $\mathcal{M}_g^F$. It is easy to see that this is not the case at the configuration $\boldsymbol{\theta}_q$ where a failure in joint 1 results in the robot being constrained to be on a failure plane that does not intersect $\mathcal{M}_g^F$ so that the robot would not be able to successfully finish its task. Similarly, a failure in joint 1 or joint 2 at $\boldsymbol{\theta}_p$ would result in failure planes that do not intersect $\mathcal{M}_g^F$. Hence configurations $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_q$ do not have the same failure tolerance properties as $\boldsymbol{\theta}_s$.

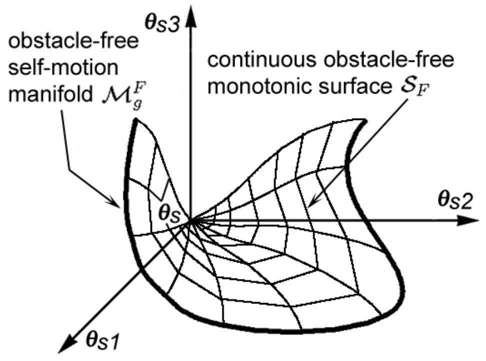
Thus to ensure the ability to reach the goal self-motion manifold, any start configuration should have the same properties as configuration $\boldsymbol{\theta}_s$ shown in Fig. 1. We state this condition formally as:

Necessary Condition. A given obstacle-free configuration $\boldsymbol{\theta}_s$ is called a feasible configuration if all the corresponding failure hyperplanes for $\boldsymbol{\theta}_s$ intersect a portion of the obstacle-free goal self-motion manifold $\mathcal{M}_g^F$, that is,

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

where $\mathbf{H}_i$ is the failure hyperplane at $\boldsymbol{\theta}_s$ corresponding to joint i. This creates a guaranteed path to $\mathcal{M}_g^F$ ensuring a possibility of successfully reaching the goal despite a joint failure at the start configuration $\boldsymbol{\theta}_s$. Note that this is equivalent to the goal position being in the fault tolerant workspace [30]. At this point, we have not considered the issue of encountering obstacles along the path from $\boldsymbol{\theta}_s$ to $\mathcal{M}_g^F$ so (6) is only a necessary condition. However, we will use this condition as a preliminary check to eliminate non-feasible solutions.

If subsequent points taken along the path towards the goal are obstacle-free and have their corresponding



θ_{s3}
obstacle-free self-motion manifold $\mathcal{M}_g^F$
continuous obstacle-free monotonic surface $\mathcal{S}_F$
θ_{s2}
θ_s
θ_{s1}

Fig. 2. A failure surface $\mathcal{S}_F$ for a 3-D C-space corresponding to an obstacle-free start configuration $\boldsymbol{\theta}_s$ (shown at the origin) and an obstacle-free portion of a goal self-motion manifold $\mathcal{M}_g^F$. Assuming that the bounds of the failure hypercube $\mathbf{V}_{sg}$ are known, $\mathcal{S}_F$ is then created by generating monotonic paths within the failure hypercube and connecting $\boldsymbol{\theta}_s$ to points in $\mathcal{M}_g^F$ via an obstacle avoidance algorithm. Obstacle-free space continuity between paths is checked by connecting the adjacent paths via straight lines.

failure planes intersect the obstacle-free goal self-motion manifold, then it is guaranteed that there is a possibility of successfully moving from the start position $\mathbf{x}_s$ to the goal position $\mathbf{x}_g$ no matter which joint fails at any given time. This set of points lies on a continuous obstacle-free monotonic surface within the failure hypercube corresponding to the start configuration $\boldsymbol{\theta}_s$. Thus, if a continuous obstacle-free monotonic surface within the failure hypercube exists, then successfully moving from a start configuration $\boldsymbol{\theta}_s$ to a point in $\mathcal{M}_g^F$ is guaranteed despite any single joint failure and despite obstacles. We formally state this condition as:

Sufficient Condition. Consider a given failure hypercube $\mathbf{V}_{sg}$ containing an obstacle-free start configuration $\boldsymbol{\theta}_s$ and an obstacle-free portion of the goal self-motion manifold $\mathcal{M}_g^F$. If $\mathcal{S}_F$ is a continuous obstacle-free monotonic surface in $\mathbf{V}_{sg}$ such that the two conditions

$$\boldsymbol{\theta}_s \in \mathcal{S}_F \text{ and } \mathcal{M}_g^F \subset \mathcal{S}_F \tag{7}$$

hold, then an obstacle-free path to the goal is guaranteed even if any single locked joint failure occurs. This serves as the final check for the existence of a solution. If this holds, a solution to the problem is guaranteed.

A failure surface is identified by generating monotonic paths, within the bounds of the failure hypercube, that connect $\boldsymbol{\theta}_s$ to points along $\mathcal{M}_g^F$ via an obstacle avoidance algorithm. Adjacent paths can be connected via straight lines to check for obstacle-free space continuity between paths. If the path generations and the path connections are collision-free, then a web of obstacle-free paths is created which represents the failure surface $\mathcal{S}_F$. Fig. 2 shows a failure surface $\mathcal{S}_F$ for a 3-DOF C-space.

As long as the robot moves along this set of points on the surface, then it is guaranteed that it would reach the desired goal position for any single joint failure despite obstacles in the workspace.

For kinematic redundancy of two or more, the self-motion manifold is a hypersurface. The algorithm would first choose a curve along the hypersurface as its goal self-motion manifold. One way of choosing this curve is by using an obstacle avoidance algorithm to identify an obstacle-free continuous curve [35], [36]. Other desired posture optimization criteria could also be used. Once a hypercurve is chosen, the algorithm would proceed in the same manner as in the case of a single degree of redundancy.

IV. The Algorithm for Failure Tolerant Motion Planning

The steps for ensuring obstacle-free failure tolerant paths are as follows:

1) Given the workspace start position, $\mathbf{x}_s$, and the goal position, $\mathbf{x}_g$, transform the given workspace positions into their equivalent C-space self-motion manifolds: start self-motion manifold, $\mathcal{M}_s$, and goal self-motion manifold, $\mathcal{M}_g$, respectively.
2) Choose configurations from $\mathcal{M}_s$ and identify those that are obstacle-free. Record these configurations and label them as $\mathcal{M}_s^F$. Similarly, the obstacle-free portion of the goal self-motion manifold is denoted as $\mathcal{M}_g^F$.
3) For each configuration in $\mathcal{M}_s^F$, the corresponding failure hyperplanes are checked against intersections with the goal self-motion manifold $\mathcal{M}_g$. If all the corresponding hyperplanes for configurations in $\mathcal{M}_s^F$ intersect $\mathcal{M}_g^F$, record all the intersections to form a subset $\boldsymbol{\Theta}_{\mathbf{H}}$ of $\mathcal{M}_g^F$. Each configuration in $\mathcal{M}_s^F$ that has a corresponding $\boldsymbol{\theta}_{\mathbf{H}} \in \boldsymbol{\Theta}_{\mathbf{H}}$ is saved to form a subset $\boldsymbol{\Theta}_s$ of $\mathcal{M}_s^F$. (This step uses the necessary condition in Section III.)
4) Identify the bounds of the failure hypercube $\mathbf{V}_{sg}$ associated with each configuration $\boldsymbol{\theta}_s \in \boldsymbol{\Theta}_s$.
5) For each failure hypercube $\mathbf{V}_{sg}$, check for the existence of a failure surface $\mathcal{S}_F$. This is done by using an obstacle avoidance algorithm to generate monotonic paths from a start configuration $\boldsymbol{\theta}_s$ to points in $\mathcal{M}_g^F$. Straight line connections between paths are used to check for the continuity of the obstacle-free space between paths.
6) If the path generations and straight line connections between paths are obstacle-free, the result is a continuous web of paths representing the failure surface $\mathcal{S}_F$ corresponding to $\mathbf{V}_{sg}$. If no $\mathcal{S}_F$ exists for any failure hypercube $\mathbf{V}_{sg}$, 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. (This step uses the sufficient condition in Section III.)

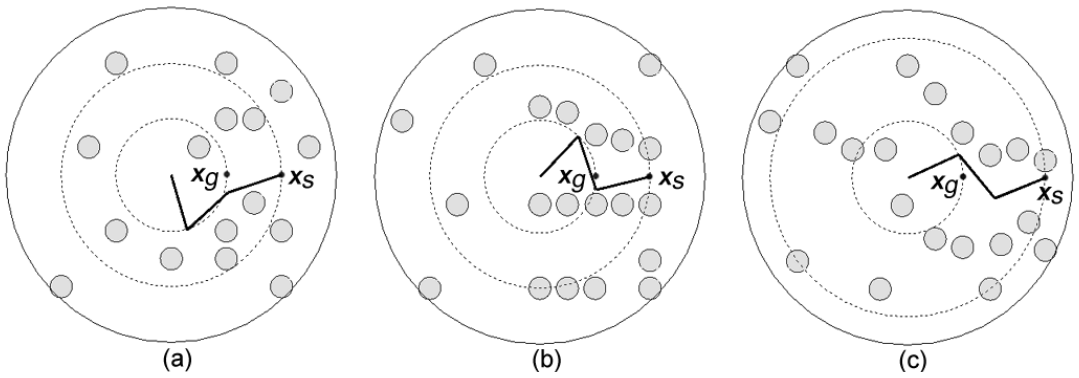


Fig. 3. A 3-DOF planar robot with three distinct sets of obstacles in the environment. For each scenario, the robot configuration shown is the feasible start configuration $\boldsymbol{\theta}_s$ for which a failure surface $\boldsymbol{\mathcal{S}}_F$ is found. Scenarios (a) and (b) have $\mathbf{x}_s$ at $[200, 0]^T$ and $\mathbf{x}_g$ at $[100, 0]^T$. Scenario (c) has $\mathbf{x}_s$ at $[250, 0]^T$ and $\mathbf{x}_g$ at $[100, 0]^T$. The robot has equal link lengths of 100 units and each obstacle has a diameter of 40 units. In all cases the robot can reach the desired position $\mathbf{x}_g$ from the shown start configuration $\boldsymbol{\theta}_s$ despite any single joint failure.

TABLE I

THE SET OF START CONFIGURATIONS SHOWN IN FIG. 3 WITH THE CORRESPONDING COORDINATES OF THE FAILURE PLANES THAT INTERSECT THE OBSTACLE-FREE GOAL SELF-MOTION MANIFOLD.

Fig. 3	Start Configuration			Failure Plane Intersection with Obstacle-Free Goal Self-Motion Manifold $\boldsymbol{\mathcal{M}}_g^F$								
	θ_{s1}	θ_{s2}	θ_{s3}	$\theta_{\mathbf{H}_1 1}$	$\theta_{\mathbf{H}_1 2}$	$\theta_{\mathbf{H}_1 3}$	$\theta_{\mathbf{H}_2 1}$	$\theta_{\mathbf{H}_2 2}$	$\theta_{\mathbf{H}_2 3}$	$\theta_{\mathbf{H}_3 1}$	$\theta_{\mathbf{H}_3 2}$	$\theta_{\mathbf{H}_3 3}$
(a)	-73.3	114.0	-23.0	-73.3	179.4	-105.9	-113.4	114.0	65.8	-154.0	178.6	-23.0
(b)	66.7	-119.1	45.1	66.7	-179.4	112.8	118.7	-119.1	-60.2	133.3	-179.2	45.1
(c)	23.5	-73.0	70.9	23.5	-178.6	155.3	-0.3	-73.0	179.4	108.9	-179.4	70.9

7) Given that an $\boldsymbol{\mathcal{S}}_F$ exists, to move from the start position, $\mathbf{x}_s$, towards the goal position, $\mathbf{x}_g$, the manipulator has to move from start configuration $\boldsymbol{\theta}_s$ along the continuous web of paths that represents the failure surface $\boldsymbol{\mathcal{S}}_F$ toward $\boldsymbol{\mathcal{M}}_g^F$. Because $\boldsymbol{\mathcal{S}}_F$ is known to be collision-free, as long as the manipulator moves along the surface it would be free from collision and at the same time it can reach the goal from any failure hyperplane.

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 a single degree of redundancy, 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 start and goal self-motion manifolds, while the second term is the contribution due to collision detection.

V. A 3-LINK PLANAR ROBOT EXAMPLE

An implementation of the path planning algorithm discussed here is shown for a 3-DOF planar robot. The robot has equal link lengths of 100 units and is required to move from a start position $\mathbf{x}_s$ to a goal position $\mathbf{x}_g$ in the workspace as shown in Fig. 3. Each disk obstacle has a diameter of 40 units. In Figs. 3(a) and 3(b), the start position is $\mathbf{x}_s = [200, 0]^T$ while in Fig. 3(c) the start position is $\mathbf{x}_s = [250, 0]^T$. In all cases the goal position is $\mathbf{x}_g = [100, 0]^T$.

The obstacles were selected in such a way that the robot will have difficulty finding a failure surface $\boldsymbol{\mathcal{S}}_F$ for a given scenario, i.e., a given $\mathbf{x}_s$, $\mathbf{x}_g$, and workspace [37]. Obstacles were added until eventually, the robot was left with a single feasible start configuration $\boldsymbol{\theta}_s$ where a failure surface $\boldsymbol{\mathcal{S}}_F$ exists. For each scenario in Fig. 3, the robot configuration shown is the feasible start configuration $\boldsymbol{\theta}_s$ that has a corresponding $\boldsymbol{\mathcal{S}}_F$.

Table I shows the values of the start configuration $\boldsymbol{\theta}_s = [\theta_{s1}, \theta_{s2}, \theta_{s3}]^T$ corresponding to each scenario in Fig. 3. The planes $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}_3$ are the corresponding failure planes that intersect the obstacle-free goal self-motion manifold $\boldsymbol{\mathcal{M}}_g^F$ at $\boldsymbol{\theta}_{\mathbf{H}_i} = [\theta_{\mathbf{H}_i 1}, \theta_{\mathbf{H}_i 2}, \theta_{\mathbf{H}_i 3}]^T$ where $i = 1, 2, 3$.

Obstacles in scenario Fig. 3(a) were chosen in a more random manner such that the robot "squeezed its way through" to find a feasible start configuration $\boldsymbol{\theta}_s$ where a corresponding $\boldsymbol{\mathcal{S}}_F$ exists. Obstacles in scenario Fig. 3(b) were chosen in such a way that they will have a more linear arrangement such that the robot is in a "walled" or "pipe-like" environment. Obstacles in scenario Fig. 3(c) were chosen randomly as in Fig. 3(a), however, the start configuration is further away from the robot's base. Notice that the most difficult obstacles in Fig. 3(c) are wider apart

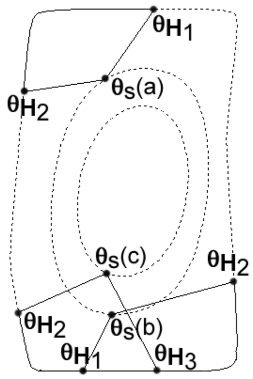


Fig. 4. A projection of the self-motion manifolds, drawn in broken lines, for $\mathbf{x}_g = [100, 0]^T$ (outermost oblong), $\mathbf{x}_s = [200, 0]^T$ (middle oblong), and $\mathbf{x}_s = [250, 0]^T$ (innermost oblong). The start configuration $\boldsymbol{\theta}_s$ that corresponds to each scenario in Fig. 3 and their corresponding failure plane intersections with the obstacle-free goal self-motion manifold $\boldsymbol{\mathcal{M}}_g^F$ are shown. The surface bounded by solid lines represent the failure surface $\boldsymbol{\mathcal{S}}_F$ for each $\boldsymbol{\theta}_s$.

from each other compared to the most difficult obstacles in Fig. 3(a). The reason is because in Fig. 3(c), the robot links are more stretched out at the start position $\mathbf{x}_s$, and if a joint failure occurs at the start configuration, the robot would need a bigger obstacle-free sweep area for it to reach the goal position $\mathbf{x}_g$.

Fig. 4 shows a projection of the C-space self-motion manifolds, drawn in broken lines, corresponding to workspace positions in Fig. 3. The goal position $\mathbf{x}_g = [100, 0]^T$ corresponds to the outermost oblong self-motion manifold, $\mathbf{x}_s = [200, 0]^T$ corresponds to the center oblong self-motion manifold, and $\mathbf{x}_s = [250, 0]^T$ corresponds to the innermost oblong self-motion manifold. Also shown are the start configurations $\boldsymbol{\theta}_s$ for the each scenario in Fig. 3 and the corresponding failure plane intersection $\boldsymbol{\theta}_{\mathbf{H}_i}$ with the obstacle-free goal self-motion manifold $\boldsymbol{\mathcal{M}}_g^F$. Notice that the straight lines connecting the start configuration $\boldsymbol{\theta}_s$ to $\boldsymbol{\theta}_{\mathbf{H}_i}$ are along the failure planes corresponding to $\boldsymbol{\theta}_s$. The failure planes are orthogonal to each other but do not appear orthogonal in the figure because the projection is taken at an angle. The failure surface $\boldsymbol{\mathcal{S}}_F$ for each $\boldsymbol{\theta}_s$ is the surface bounded by the solid lines. Indeed, as shown in Fig. 4, a failure surface $\boldsymbol{\mathcal{S}}_F$ exists that would guarantee successfully reaching the goal despite obstacles in the environment and despite any single joint failure at any time.

VI. Conclusion

The problem of guaranteeing failure tolerant operation of a kinematically redundant manipulator in an environment containing obstacles was considered. In particular, conditions were determined which guarantee that a manipulator could successfully reach a goal position in the workspace from a given start position despite any single locked-joint failure and despite the presence of obstacles in a static environment. Based on these conditions an algorithm was introduced that would search for a continuous obstacle-free monotonic surface between an obstacle-free start configuration and an obstacle-free goal self-motion manifold in C-space that would determine the existence of a failure tolerant path for a given robot and environment. The method would help in deciding whether to proceed or reposition a robot so that it is guaranteed that a critical task can be completed in the presence of obstacles.[2]

VII. References

[1] E. C. Wu, J. C. Hwang, and J. T. Chladek, “Fault-tolerant joint development for the space shuttle remote manipulator system: Analysis and experiment,” *IEEE Trans. Robot. Automat.*, vol. 9, no. 5, pp. 675–684, Oct. 1993.

[2] G. Visentin and F. Didot, “Testing space robotics on the Japanese ETS-VII satellite,” ESA Bulletin-European Space Agency, 1999.

[3] P. S. Babcock and J. J. Zinchuk, “Fault-tolerant design optimization: Application to an autonomous underwater vehicle navigation system,” in *Proc. 1990 Symp. Autonom. Underwater Vehicle Technol.*, Washington, D.C., June 5-6 1990, pp. 34–43.

[4] R. Colbaugh and M. Jamshidi, “Robot manipulator control for hazardous waste-handling applications,” *J. Robot. Syst.*, vol. 9, no. 2, pp. 215–250, 1992.

[5] W. H. McCulloch, “Safety analysis requirements for robotic systems in DOE nuclear facilities,” in *Proc. 2nd Specialty Conf. Robot. Challenging Environ.*, Albuquerque, NM, June 1-6 1996, pp. 235–240.

[6] M. L. Leuschen, I. D. Walker, and J. R. Cavallaro, “Investigation of reliability of hydraulic robots for hazardous environment using analytic redundancy,” in *Proc. Annual Rel. Maintain. Symp.*, Washington, D.C., Jan. 18-21 1999, pp. 122–128.

[7] B. S. Dhillon, *Robot Reliability and Safety*. New York: Springer-Verlag, 1991.

[8] J. F. Engelberger, “Three million hours of robot field experience,” *Ind. Robot*, pp. 164–168, June 1974.

[9] B. S. Dhillon and A. R. M. Fashandi, “Safety and reliability assessment techniques in robotics,” *Robotica*, vol. 15, no. 6, pp. 701–708, Nov.-Dec. 1997.

[10] K. Khodabandehloo, “Analysis of robot systems using fault and event trees: Case studies,” *Rel. Eng. Syst. Safety*, vol. 53, no. 3, pp. 247–264, Sept. 1996.

[2]The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

[11] I. D. Walker and J. R. Cavallaro, “The use of fault trees for the design of robots for hazardous environments,” in *Proc. Annual Rel. Maintain. Symp.*, Las Vegas, NV, Jan. 22-25 1996, pp. 229–235.

[12] C. Carreras and I. D. Walker, “Interval methods for fault-tree analysis in robotics,” *IEEE Trans. Robot. Automat.*, vol. 50, no. 1, pp. 3–11, March 2001.

[13] J. Lee, “Measurement of machine performance degradation using a neural network model,” *Comput. Ind.*, vol. 30, no. 3, pp. 193–209, Oct. 1996.

[14] S. Tosunoglu and V. Monteverde, “Kinematic and structural design assessment of faul-tolerant manipulators,” *Intell. Automat. Soft Comput.*, vol. 4, no. 3, pp. 261–268, 1998.

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

[16] 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.

[17] D. L. Schneider, D. Tesar, and J. W. Barnes, “Development & testing of a reliability performance index for modular robotic systems,” in *Proc. Annual Rel. Maintain. Symp.*, Anaheim, CA, Jan. 24-27 1994, pp. 263–271.

[18] D. L. Hamilton, I. D. Walker, and J. K. Bennett, “Fault tolerance versus performance metrics for robot systems,” *Rel. Eng. Syst. Safety*, vol. 53, pp. 309–318, 1996.

[19] B. S. Dhillon and N. Yang, “Formulas for analyzing a redundant robot configuration with a built-in safety system,” *Microelectron. Rel.*, vol. 37, no. 4, pp. 557–563, April 1997.

[20] W. S. Ng and C. K. Tan, “On safety enhancements for medical robots,” *Rel. Eng. Syst. Safety*, vol. 54, no. 1, pp. 35–35, Oct. 1996.

[21] J. Trevelyan, “Simplifying robotics - a challenge for research,” *Robot. Autonom. Syst.*, vol. 21, no. 3, pp. 207–220, Sept. 1997.

[22] M. Savsar, “Reliability analysis of a flexible manufacturing cell,” *Rel. Eng. Syst. Safety*, vol. 67, no. 2, pp. 147–152, Feb. 2000.

[23] Y. Ting, S. Tosunoglu, and D. Tesar, “A control structure for fault-tolerant operation of robotic manipulators,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Atlanta, GA, May 2-6 1993, pp. 684–690.

[24] K. S. Tso, M. Hecht, and N. I. Marzwell, “Fault-tolerant robotic system for critical applications,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Atlanta, GA, May 2-6 1993, pp. 691–696.

[25] 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.

[26] T. S. Wikman, M. S. Branicky, and W. S. Newman, “Reflexive collision avoidance: A generalized approach,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Atlanta, GA, May 2-6 1993, pp. 31–36.

[27] 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.

[28] S. K. Ralph and D. K. Pai, “Computing fault tolerant motions for a robot manipulator,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, Detroit, MI, May 10-15 1999, pp. 486–493.

[29] 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.

[30] 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.

[31] 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.

[32] C. L. Lewis and A. A. Maciejewski, “Dexterity optimization of kinematically redundant manipulators in the presence of joint failures,” *Comput. Electr. Eng.*, vol. 20, no. 3, pp. 273–288, 1994.

[33] 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.

[34] J. W. Burdick, “On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, Scottsdale, AZ, May 14-19 1989, pp. 264–270.

[35] 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.

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

[37] 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.