1993

Kinematic Design of Serial Link Manipulators from Task Specifications

Christiaan J.J. Paredis
Carnegie Mellon University

Pradeep Khosla
Carnegie Mellon University

Christiaan J.J. Paredis and Pradeep K. Khosla,
"Designing Fault Tolerant Manipulators: How Many Degrees-of-freedom?"
to *appear in The International Journal of Robotics Research,* 1996.

Designing Fault Tolerant Manipulators: How Many Degrees-of-Freedom?

Christiaan J. J. Paredis and Pradeep K. Khosla

Department of Electrical and Computer Engineering and
The Robotics Institute,
Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213.

Abstract

One of the most important parameters to consider when designing a manipulator is the number of degrees-of-freedom (DOFs). This article focuses on the question: How many DOFs are necessary and sufficient for fault tolerance and how should these DOFs be distributed along the length of the manipulator? A manipulator is fault tolerant if it can complete its task even when one of its joints fails and is immobilized. The number of degrees-of-freedom needed for fault tolerance strongly depends on the knowledge available about the task. In this article, two approaches are explored. First, for the design of a General Purpose Fault Tolerant Manipulator, *it is assumed that neither the exact task trajectory, nor the redundancy resolution algorithm are known a priori and that the manipulator has no joint limits. In this case, two redundant DOFs are necessary and sufficient to sustain one joint failure as is demonstrated in two design templates for spatial fault tolerant manipulators. In a second approach, both the Cartesian task path and the redundancy resolution algorithm are assumed to be known. The design of such a* Task Specific Fault Tolerant Manipulator *requires only one degree-of-redundancy.*

1 Introduction

As robots are being used in a growing range of applications, the issue of reliability becomes more and more important. Recently, with the Hubble telescope and the Mars Observer, NASA has experienced first hand how devastating the consequences can be when a critical component fails during a multi-billion-dollar mission. Space applications are particularly vulnerable to failure, because of the adverse environment (cosmic rays, solar particles etc.) and the demand for long term operation. In this context, NASA has started to incorporate fault tolerance in their robot designs (Wu et al. 1993). Reliability is also important in medical robotics, because of the risk of the loss of human life. Although medical staff will probably always be on standby to take over in the case of a manipulator failure, the robot should at least be fail-safe, meaning it should fail into a safe configuration. A third domain of robot applications in which reliability is a major issue is the Environmental Restoration and Waste Management (ER&WM) program of the Department of Energy. Consider, for instance, the use of a manipulator in a nuclear environment where equipment has to be repaired or space has to be searched for radioactive contamination. The manipulator system deployed in these kinds of critical tasks must be reliable, so that the successful completion of the task or the safe removal of the robot system is assured.

In this article, we focus on fault tolerance as a technique to achieve reliability in manipulator systems. The traditional approach to reliability has been that of fault intolerance, where the reliability of the system is assured by the use of high quality components. However, increasing system complexity and the necessity for long term operation have proven this approach inadequate. The system reliability can be further improved through redundancy. This design approach was already advocated in the early fifties by von Neumann in connection with the design of reliable computers: "The complete system must be organized in such a manner, that a malfunction of the whole automaton cannot be caused by the malfunctioning of a single component, ... , but only by the malfunctioning of a large number of them" (von Neumann 1956, p. 70). This is the basic principle of fault tolerance: *add redundancy to compensate for possible failures of components*. However, this does not mean that any kind of redundancy added to a system results in fault tolerance. The main goal of this article is therefore to shed some light on the redundancy requirements for fault tolerant manipulators. That is, how much redundancy is needed and how should this redundancy be distributed over the manipulator structure?

The redundancy provisions needed for fault tolerance can be incorporated only at a price of increased complexity. This drawback can be overcome by a modular and structured design philosophy, as is advocated in (Schmitz, Khosla, and Kanade 1988; Fukuda et al. 1992; Sreevijayan 1992; Hui et al. 1993; Chen and Burdick 1995). Modularity in hardware and software has the advantage of facilitating testing during the design phase and therefore reducing the chances for unanticipated faults. Modules also constitute natural boundaries to which faults can be confined. By including fault detection and recovery mechanisms in critical modules, the effect of local faults remains internal to the modules, totally transparent to the higher levels of the manipulator system. Such a modular design philosophy is embodied in the Reconfigurable Modular Manipulator System (RMMS) developed in the Advanced Manipulators Laboratory at Carnegie Mellon University (Schmitz, Khosla, and Kanade 1988). The

RMMS utilizes a stock of interchangeable link modules of different lengths and shapes, and joint modules of various sizes and performance specifications. By combining these general purpose modules, a wide range of manipulator configurations can be assembled. When one needs a different configurations for a specific task (Paredis, and Khosla 1993), a robot created with the RMMS can be easily taken apart and reconfigured suitably. This reconfigurability can be further exploited to reduce the complexity of fault tolerant manipulators, as is shown in Section 4.

Over the past decade, a lot of research has been done in fault tolerance for computer systems (refer to Johnson (1989) for and overview), but only recently has the concept been applied in robotics. Most of the work in fault tolerant robotics is directly based on the results from computer science, and can be classified in three categories:

1. Design of fault tolerant robots,
2. Fault detection and identification (FDI),
3. Fault recovery and intelligent control.

When designing a fault tolerant manipulator, one should decide where to include redundancy so that the overall reliability is maximum. One should distinguish between hardware, software, analytical, information, and time redundancy (Johnson 1989). Our focus will be on hardware redundancy, which consists of actuation, sensor, communication and computing redundancy. Each of these types of redundancies can still be implemented at different levels. In Sreevijayan (1992), for instance, a four-level subsumptive architecture for actuation redundancy is proposed:

Level 1: Dual actuators—extra actuators per joint,

Level 2: Parallel structures—extra joints per DOF,

Level 3: Redundant manipulators—extra DOFs per manipulator arm,

Level 4: Multiple arms—extra arms per manipulator system.

A system can possibly be designed with redundancies at all four levels, resulting in the ability to sustain multiple simultaneous faults.

An example of a fault tolerant design for the space shuttle manipulator is described in Wu et al. (1993). Fault tolerance is here guaranteed by using a differential gear train with dual input actuators for every DOF—an implementation of the first level of the four-level subsumptive architecture. In this article, we are more interested in achieving fault tolerance using redundant DOFs (Level 3). We envision the following scenario. A fault detection and identification algorithm monitors the proper functioning of each DOF of a redundant manipulator. As soon as a failure of a subcomponent is detected, one immobilizes the corresponding DOF by activating its fail-safe brake. Automatically, the joint trajectory is adapted to the new manipulator structure and the task is continued without interruption. The strength of this scenario resides in the fact that it can handle a large variety of possible faults, ranging from sensor failures to transmission and actuation failures. All these failures can be treated in the same manner, namely, by eliminating the whole DOF through immobilization.

Although fault detection and identification (FDI) is an important part of our scenario for fault tolerance, we will not cover this subject in this article. Instead we refer to the following references (Chow and Willsky 1984; Stengel 1988; Ting, Tosunoglu, and Tesar 1993; Visinsky, Walker, and Cavallaro 1993; Visinsky, Walker, and Cavallaro 1994). In Visinsky, Walker, and Cavallaro (1993), using the concept of analytical redundancy, an FDI algorithm is presented along the lines of Chow and Willsky (1984). The result is a set of four simple equations which test for consistency between the measured position and velocity and the expected acceleration and jerk. This FDI algorithm fits into a three-layer intelligent control framework, consisting of a servo layer, and interface layer and a supervisory layer. The main problem presented to the intelligent controller is to distinguish between failures, disturbances and modeling errors, and to respond to each in the proper way. An overview of intelligent fault tolerant control is given in Stengel (1988). A range of approaches is reported, beginning with robust control, progressing through parallel and analytical redundancy, and ending with rule-based systems and artificial neural networks. The task of the robotics researcher is to apply and modify these approaches to the highly non-linear dynamics of robot manipulators.

After a fault has been detected, the failing DOF is immobilized by activating its brake. In Pradeep et al. (1988), the authors analyze the effect of the immobilization of one of the DOFs of three commercial manipulators. They conclude that the robots with decoupled DOFs are more severely "crippled" by the loss of a joint than the ones with strongly coupled DOFs. This can be translated into the guideline that, for the design of fault tolerant manipulators, strong coupling between the DOFs is highly desirable. The results presented in Lewis and Maciejewski (1994a) can be interpreted similarly. A kinematic fault tolerance measure is defined as the minimum kinematic dexterity after joint failure. The maximum kinematic fault tolerance is achieved in a manipulator posture in which each joint contributes equally to the null-space motion—a posture with strong coupling between the DOFs. For a manipulator with at least one decoupled DOF, the kinematic fault tolerance measure is always minimal, that is, zero. The same measure can be used as a criterion for the redundancy resolution of the fault-free manipulator. It is shown that the chances for task completion, after immobilization of one joint due to failure, are much better than when traditional pseudo-inverse control is used. However, due to the local nature of the fault tolerance measure, completion of the task cannot be guaranteed on a global scale (Lewis and Maciejewski 1994b). An important conclusion is that the ability to recover from a fault depends strongly on the joint trajectory followed by the fault-free manipulator system.

This conclusion led us to explore two approaches to the problem of manipulator fault tolerance. The two approaches differ in the assumptions that are made with regard to the task and with regard to the choice of redundancy resolution algorithm. In a first approach, the goal is to design a *general purpose fault tolerant manipulator.* We assume that the task is only characterized by the size and position of the *task space* which is the portion of the Cartesian output space in which the task will take place. No assumptions are made about the path that needs to be followed within the task space or about the redundancy resolution algorithm used to execute the task. Such a general purpose fault tolerant manipulator can fault tolerantly execute any task of which the task space lies inside the fault tolerant workspace of the manipulator. This approach to fault tolerance is further explored in Section 3.

In a second approach, the goal is to design a *task specific fault tolerant manipulator.* In this approach, we assume that the Cartesian path to be followed is known a priori and that the corresponding set of possible joint trajectories can be limited by an appropriate choice of a redundancy resolution algorithm. We show in Section 4 how these additional assumptions allow us to design a fault tolerant manipulator with fewer DOFs than a general purpose fault tolerant manipulator.

For both approaches, we will in this article answer the question: How many degrees-of-redundancy (DORs) are necessary and sufficient for fault tolerance?

2 Fault Tolerance and Reliability

The basic idea presented in this article is to use a manipulator's redundant DOFs to compensate for a possible failure of one of the joints. The underlying assumption is that a manipulator that can sustain a joint failure is more reliable than one that cannot. The question is: "Does fault tolerance always result in an increase in reliability?" The answer is given by reliability theory (Johnson 1989).

The reliability, $R(t)$, of a component or a system is the conditional probability that the component operates correctly throughout the interval $[t_0, t]$, given that it was operating correctly at the time t_0. For non-fault-tolerant serial link manipulators, the system fails when any single subsystem—a DOF or joint module for modular manipulators—fails. The system reliability can then be computed as the product of the module reliabilities, $R_i(t)$:

$$R_s(t) = R_1(t)R_2(t)\dots R_n(t). \tag{1}$$

Or, in the case that every module is equally reliable with reliability $R_{mod}(t)$:

$$R_s(t) = R^n_{mod}(t). \tag{2}$$

If there are n modules and only m of those are required for the system to function properly—the system can tolerate $(n-m)$ module failures—then the system reliability is the sum of the reliabilities of all systems with $(n-m)$ or fewer faults. Since there are $\binom{n}{i}$ different systems with i faults, the system reliability of a fault tolerant system with equal module reliabilities can be written concisely as

$$R_s(t) = \sum_{i=0}^{n-m} \binom{n}{i} R^{n-i}_{mod}(t)(1-R_{mod}(t))^i . \tag{3}$$

We can apply this formula to the example of an 8-DOF fault tolerant manipulator, which needs only seven DOFs to function properly. The system reliability of the fault tolerant system is:

$${}^fR_s(t) = R^8_{mod}(t) + 8R^7_{mod}(t)(1-R_{mod}(t)) = R^7_{mod}(t)(8-7R_{mod}(t)) , \tag{4}$$

compared to $R_s(t) = R^6_{mod}(t)$ for an equivalent 6-DOF non-fault-tolerant system. Both reliabilities are plotted as a function of the module reliability in Figure 1a, while Figure 1b shows the relative system reliability ${}^fR_s / R_s$:

$$\frac{{}^fR_s}{R_s} = R_{mod}(8-7R_{mod}), \tag{5}$$

which equals 1 for $R_{mod} = 1$ and $R_{mod} = \frac{1}{7}$. These graphs should be interpreted as follows:

- when $R_{mod}(t) = 0$, then ${}^fR_s = R_s = 0$. The system reliability is zero in both cases, i.e., both systems are guaranteed to fail.
- when $R_{mod}(t) = 1$, then ${}^fR_s = R_s = 1$. That is, both systems are 100% reliable.
- when $\frac{1}{7} < R_{mod}(t) < 1$, then ${}^fR_s > R_s$, meaning that the fault tolerant system is more reliable than the non-fault-tolerant one.

- when $R_{mod}(t) < \frac{1}{7}$, then ${}^{f}R_s < R_s$. The modules are so unreliable that the added complexity of the fault tolerant system reduces the overall performance.

Preferably, one would like to operate a system at a reliability close to one, for which the fault tolerant system is the more reliable. To compare both alternatives for modules with a high reliability, it is more instructive to rewrite Equation (4) as an expression for the system's *unreliability*, $Q(t) = 1 - R(t)$:

$$ {}^{f}Q_s(t) \approx 28Q_{mod}^2(t) - 112Q_{mod}^3(t), \tag{6}$$

when $Q_{mod} \to 0$. The unreliability for the non-fault-tolerant system is

$$Q_s(t) \approx 6Q_{mod}(t) - 15Q_{mod}^2(t). \tag{7}$$

In general, the unreliability of a k-fault tolerant system—one that can sustain k faults—is of the order $O(Q_{mod}^{k+1})$. This means that the reliability of a fault tolerant system increases more significantly when the reliability of the individual modules is high. Best results are thus obtained when fault tolerance is combined with high component reliability, or fault intolerance.

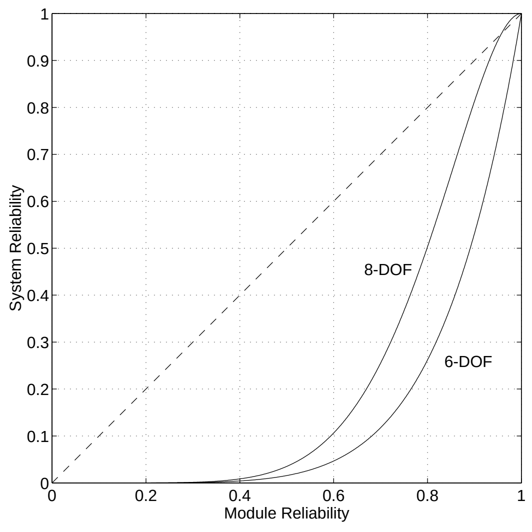

Figure 1a: System reliability of an 8-DOF fault tolerant manipulator and a 6-DOF non-fault-tolerant manipulator.

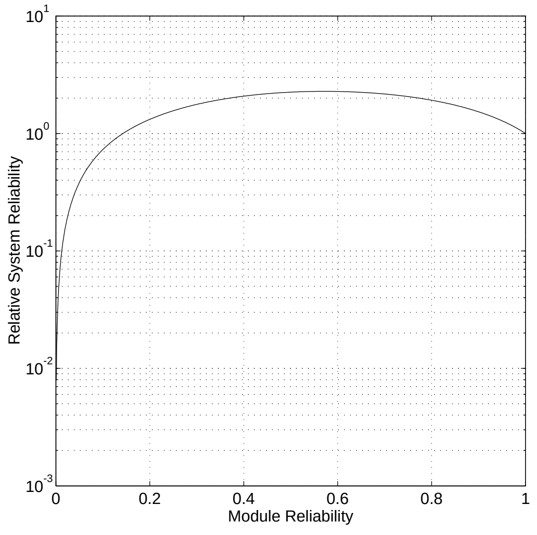

Figure 1b: Relative system reliability of an 8-DOF fault tolerant manipulator versus a 6-DOF non-fault-tolerant manipulator.

3 General Purpose Fault Tolerant Manipulators[1]

In this section, we discuss the kinematic design of a general purpose fault tolerant manipulator without joint limits. As for non-fault-tolerant manipulators, the kinematic capabilities are mainly characterized by the shape and size of the workspace or rather the fault tolerant workspace (FTWS) in this case. We identify several properties of general purpose fault tolerant manipulators and their workspaces, and propose an 8-DOF design template.

To set the stage for our development, we define the following concepts relating to general purpose fault tolerant manipulators:

- **General Purpose Fault Tolerant Manipulator**: A manipulator that will still be able to meet the task specifications, even if any one or more of its joints fail and are frozen at any arbitrary joint angles.

- ***k*-Reduced Order Derivative (*k*-ROD)**: When k joints of an n-DOF manipulator fail, the effective number of joints is $(n-k)$. The resulting faulty manipulator is called a k-reduced order derivative.

- **Order of Fault Tolerance**: A manipulator is fault tolerant of the k-th order, if and only if all possible k-reduced order derivatives can still perform the specified task. We call the manipulator k-fault tolerant.

- **Fault Tolerant Workspace (FTWS)**: The fault tolerant workspace of a k-fault tolerant manipulator is the set of points reachable by all possible k-reduced order derivatives.

Notice that our definition of a general purpose fault tolerant manipulator reflects the assumption that the redundancy resolution algorithm is not known a priori: a joint failure can occur at an arbitrary angle.

In the remainder of this section, if no specific task is mentioned, it is assumed that the task is to reach a nonzero volume of points. That is, the task space is an m-dimensional manifold in the m-dimensional output space of the manipulator. A manipulator with a FTWS of dimension less than m is considered not to be fault tolerant.

1. This section is based on Paredis and Khosla (1994).

3.1 Properties of General Purpose Fault Tolerant Manipulators

3.1.1 Existence

A general purpose manipulator has six DOFs which allow it to position its end effector in an arbitrary position and orientation anywhere in its workspace. An obvious way to make this manipulator fault tolerant is to design every joint with a redundant actuator. If one of the actuators of the resulting $2n$-DOF fault tolerant manipulator were to fail, the redundant actuator could take over and the manipulator would still be functional. Similarly, a k-fault tolerant manipulator can be constructed by duplicating every DOF k times, resulting in a $(k+1)n$ -DOF manipulator. This argument illustrates that $(k+1)n$ DOFs are sufficient for k-th order fault tolerance. In the remainder of this section, we determine the number of DOFs *necessary* to achieve general purpose fault tolerance.

3.1.2 Boundary of the Fault Tolerant Workspace

In this section, we show that a boundary point of the FTWS is a critical value.[2] Consider a k-fault tolerant planar manipulator, M. A boundary point, $\boldsymbol{p}_b$, of the FTWS has to be an element of the boundary of the workspace of at least one ROD, M^*, obtained by freezing k joints of M. Indeed, if $\boldsymbol{p}_b$ were an interior point of the workspaces of all RODs, then it would by definition be an interior point of the FTWS and not a boundary point. The Jacobian of M^*, J_{M^*}, can be obtained from the Jacobian of M, J_M, by deleting the columns corresponding to the frozen DOFs. Because $\boldsymbol{p}_b$ is a boundary point of the workspace of M^*, the Jacobian of M^* at $\boldsymbol{p}_b$ is singular. We prove now that J_M is singular too. Suppose that J_M were non-singular, then at least one of the columns corresponding to a frozen DOF would be outside the column space of the singular matrix, J_{M^*}. Physically this means that a small change in the angle of that frozen DOF would cause the end effector of M to move in a direction with a component perpendicular to the boundary of the workspace of the ROD, M^*, as illustrated in Figure 2. The ROD with this new frozen angle would be unable to reach the point, $\boldsymbol{p}_b$. As a result, $\boldsymbol{p}_b$ would be outside the FTWS, contradicting the fact that $\boldsymbol{p}_b$ is a boundary point of the FTWS. Thus, J_M is singular and $\boldsymbol{p}_b$ is a critical value.

Consequently, the FTWS is bounded by critical value manifolds. For planar positional manipulators, the critical value manifolds are concentric circles, and the FTWS is an annulus with inner radius $R_{\min}^{\text{FTWS}}$ and outer radius $R_{\max}^{\text{FTWS}}$.

2. A critical value is an end-effector position that can be reached in a singular configuration, i.e., that is the image of a critical point (Burdick 1988).

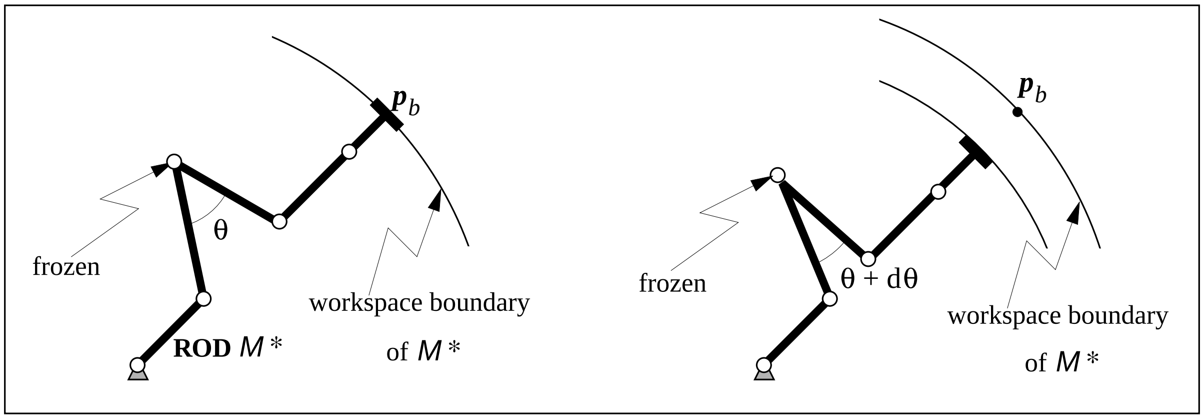

Figure 2: A ROD unable to reach a point outside the FTWS.

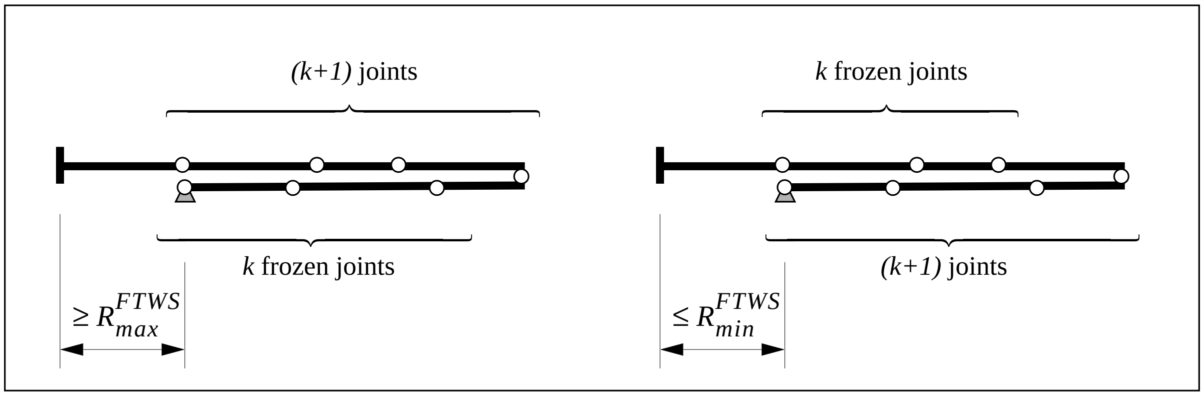

Figure 3: An upper bound for $R_{\max}^{\text{FTWS}}$ and a lower bound for $R_{\min}^{\text{FTWS}}$

3.1.3 Required Degree of Redundancy

In Section 3.1.1, it is shown that, in general, kn redundant DOFs—i.e. $(k+1)n$ DOFs in total—are sufficient to achieve k-th order fault tolerance. For planar positional manipulators, however, we prove that $2k$ redundant DOFs are necessary and sufficient for k-th order fault tolerance.

Necessary: The proof shows that $(2k+1)$ DOFs (or $2k-1$ redundant DOFs) are insufficient, by finding a lower bound for $R_{\min}^{\text{FTWS}}$ and an upper bound for $R_{\max}^{\text{FTWS}}$. First consider the ROD obtained by freezing the first k joints at 0 radians, as illustrated in Figure 3. The maximum reach in the opposite direction is an upper bound for $R_{\max}^{\mathrm{FTWS}}$:

$$R_{\max}^{\mathrm{FTWS}} \le -\sum_{i=1}^{k} l_i + l_{k+1} + \sum_{i=k+2}^{2k+1} l_i, \tag{8}$$

where l_i is the length of the i-th link. In order for $R_{\max}^{\mathrm{FTWS}}$ to be positive, we must have that:

$$\sum_{i=1}^{k} l_i \le l_{k+1} + \sum_{i=k+2}^{2k+1} l_i. \tag{9}$$

Making this assumption, we find that $R_{\min}^{\mathrm{FTWS}}$ is bounded below by the inner radius of the workspace of the ROD obtained by freezing the k last joints at 0 radians, as illustrated in Figure 3:

$$R_{\min}^{\mathrm{FTWS}} \ge \sum_{i=k+2}^{2k+1} l_i + l_{k+1} - \sum_{i=1}^{k} l_i. \tag{10}$$

From Equation (8) and Equation (10), it follows that at best

$$R_{\max}^{\mathrm{FTWS}} = R_{\min}^{\mathrm{FTWS}}, \tag{11}$$

resulting in a one-dimensional FTWS. Therefore, a $(2k+1)$-DOF planar manipulator cannot be k-th order fault tolerant.

Sufficient: The proof shows that there exists a $(2k+2)$-DOF manipulator template that is k-fault tolerant. Consider a manipulator with $(2k+2)$ links of length l. Because all the links have the same length, it is possible to compensate for a fault in a DOF by choosing a neighboring DOF to be at π radians; that is, folded back onto the failing DOF. Even when consecutive DOFs fail, this trace-back-mechanism can be used to compensate for failures. The result is that, by sacrificing one DOF to compensate for every fault, the $(2k+2)$-DOF manipulator with k faults is equivalent to a faultless 2-DOF manipulator. The FTWS of the $(2k+2)$ DOF manipulator is then the workspace of the equivalent 2-DOF manipulator, that is,

$$FTWS = \{(x, y) | \sqrt{x^2 + y^2} \le 2l\}. \tag{12}$$

3.1.4 Including Orientation

Thus far, we have only considered planar positional manipulators. The results can be easily extended to the case in which orientation is considered also, by converting the orientational problem into an equivalent positional problem:

An n-DOF manipulator, M, is k-fault tolerant with respect to a set of points, $W = \{(x_i, y_i, \varphi_i)\}$, if and only if:

1. The positional manipulator, M', obtained from M by deleting its last link, l_n, is k-fault tolerant with respect to the set of points $W' = \{(x_i - l_n \cos\varphi_i, y_i - l_n \sin\varphi_i)\}$,
2. M' is $(k-1)$-fault tolerant while reaching the points in W' in any direction.

Necessary: The positional manipulator, M', needs at least $(2k+2)$ DOFs to be k-th order fault tolerant with respect to W'; therefore, the manipulator M needs at least $(2k+3)$ DOFs.

Sufficient: Again, we show that there exists a $(2k+3)$-DOF manipulator template that is k-fault tolerant. Consider a template of which the first $(2k+2)$ links have length l and the last link has length zero; it is the template described in the previous section with a zero-length link added at the end. For this template, one can again use the trace-back-mechanism to show that it is equivalent to a faultless 3-DOF manipulator with link lengths l, l, and 0. the FTWS is thus:

$$FTWS = \left\{(x, y, \varphi) | \sqrt{x^2 + y^2} \le 2l \text{ and } \varphi \in [0, 2\pi)\right\} \tag{13}$$

This result for planar manipulators with orientation and the result obtained in Section 3.1.3 can be summarized in the following theorem:

Theorem:

For planar manipulators without joint limits, $2k$ degrees-of-redundancy are necessary and sufficient for k-th order general purpose fault tolerance.

3.2 Spatial Fault Tolerant Manipulators

For planar fault tolerant manipulators, we were able to prove that $2k$ is the required degree of redundancy. The proof was based on geometric workspace analysis. For spatial manipulators, however, the geometric analysis becomes too complex. Therefore, we will demonstrate some properties of spatial fault tolerant manipulators using two examples.

As a first example, consider a 5-DOF spatial positional manipulator. Its Denavit-Hartenberg (D-H) parameters are listed in Table 1. This manipulator is first order fault tolerant, and because of its simple kinematic structure, an analytic expression for the boundary of the FTWS can be derived. The FTWS is symmetric with respect to the first axis. A cross-section (the X-Z plane), as shown in Figure 4, can be described by two segments of a circle with radius 2 and center at $(x = 1, z = 0)$, and a straight line from $(x = 2, z = \sqrt{3})$ to $(x = 2, z = -\sqrt{3})$. An important property of this FTWS is that it does not have any holes or a central void, so that the FTWS of the same manipulator scaled by any factor $\lambda > 1$ contains the original FTWS. As a result, this fault tolerant manipulator can be used as a *design template*. Any task space can be enclosed in the FTWS of a scaled version of the design template.

In Section 3.1.2, it is shown that the boundary of the FTWS of a planar manipulator coincides with its critical value manifolds. Figure 4 demonstrates that this property also holds for the 5-DOF spatial manipulator considered in this example. The critical value manifolds are computed using the algorithm described in Burdick (1992) and are depicted in a solid line. The bold part of the critical value manifolds is the boundary of the FTWS. The property that the FTWS is bounded by critical value manifolds can be effectively used for the determination of the FTWS. Testing whether a point is an element of the FTWS is a complicated procedure. One has to verify whether that point is reachable for all possible RODs, i.e., for all manipulator structures resulting from a joint failure of every possible joint at every possible joint angle. To find a good approximation of the FTWS, one would have to execute this test for a large number of points. This would be prohibitively slow. However, to improve the efficiency, one can compute the critical value manifolds of the manipulator first. These manifolds partition the Cartesian output space of the manipulator in sectors that are either entirely inside the FTWS or entirely outside the

DOF i	d_i	a_i	α_i
1	0	1	90˚
2	a	1	0˚
3	-a	1	90˚
4	b	1	0˚
5	-b	1	—

Table 1: D-H parameters of a 5-DOF first order fault tolerant spatial manipulator, without orientation.

DOF i	d_i	a_i	α_i
1	0	1	90˚
2	a	1	0˚
3	-a	1	90˚
4	b	1	0˚
5	-b	0	90˚
6	1	0	90˚
7	0	0	90˚
8	0	0	—

Table 2: D-H parameters of an 8-DOF first order fault tolerant spatial manipulator.

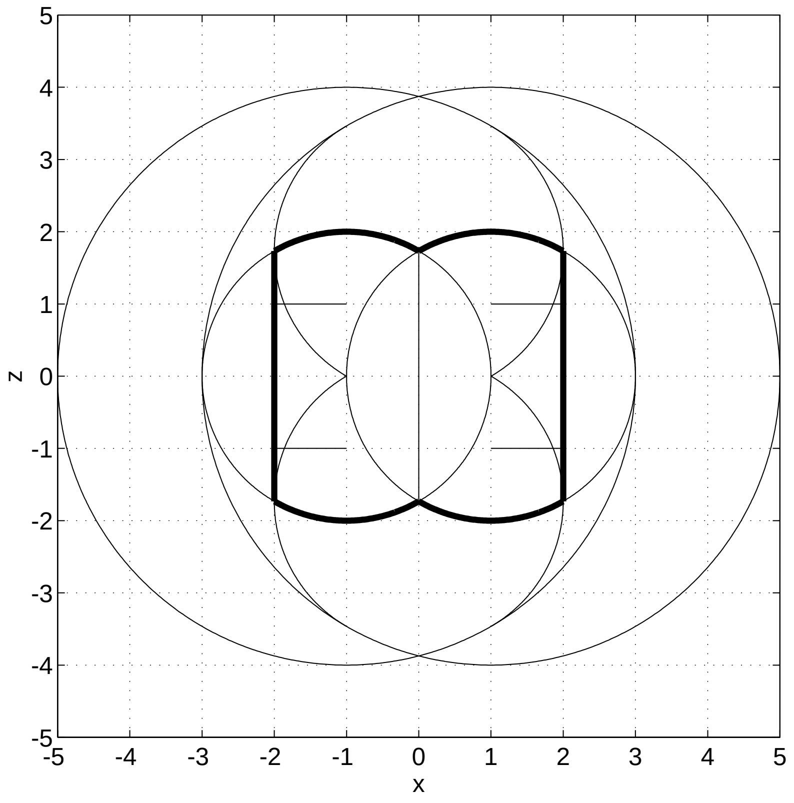

Figure 4: A cross-section of the boundary of the FTWS of a 5-DOF spatial manipulator (bold) as part of its critical value manifolds.

FTWS. Thus, by checking whether *one point* of a sector is an element of the FTWS, one can check whether the *whole sector* is in the FTWS. The number of FTWS-membership tests is so reduced to the number of sectors in the partition of the output space.

As a second example of a spatial general purpose fault tolerant manipulator, consider an 8-DOF manipulator, with D-H parameters listed in Table 2. It is the same manipulator as in the previous example, with a zero-length 3-roll-wrist added at the end. Using a Monte Carlo method, it has been determined that this manipulator is first order fault tolerant while reaching all the points, in the FTWS of example one, *in any direction*. This property can be demonstrated with the following arguments. When one of the first five DOFs fails, the manipulator can still reach any position in the FTWS (because the 5-DOF positional manipulator is FT) and can take any orientation at this position using the intact 3-roll-wrist. When one of the DOFs in the wrist fails, we are left with a 7-DOF manipulator which has enough orientational capabilities to reach any point in the FTWS in any orientation. Consequently, one could call this the *dextrous* FTWS. Since there are again no holes or voids in the FTWS, this manipulator can also be used as a design template.

Finally, one should notice that both examples have only two redundant DOFs, which indicates that two degrees-of-redundancy are also sufficient for 1-fault tolerance of spatial manipulators. Whether the theorem in Section 3.1.4 also holds for higher orders of fault tolerance of spatial manipulators requires further research.

4 Task Specific Fault Tolerant Manipulators

In the previous section, we considered the design of fault tolerant manipulators for general use. We proved that two redundant DOFs are necessary and sufficient for first order fault tolerance. However, as we will show in this section, a simpler kinematic structure is often sufficient when one specific task is considered.

The disadvantage of this approach is that a task specific fault tolerant manipulator is only suited for a very limited set of similar trajectories. Unlike general purpose fault tolerance, task specific fault tolerance might require a different manipulator structure for every task. However, by using a modular manipulator system such as RMMS, it is possible to quickly reconfigure the manipulator to custom-tailor it for a specific task.

4.1 Local versus Global Fault Tolerance

Consider the task of reaching all the points in an ε-neighborhood, $B(p, \varepsilon)$, of the point $p \in \Re^m$. Suppose that p can be reached by an n-DOF manipulator in a posture, $\theta \in T^n$. If the posture, θ, is non-singular, then there exists an $\varepsilon > 0$, such that the manipulator can reach any point in $B(p, \varepsilon)$. However, for k-fault-tolerance, any point in $B(p, \varepsilon)$ needs to be reachable even when k of the joints of the manipulator are frozen. This is possible if and only if the Jacobians of all k-RODs in the posture θ are non-singular, i.e., have at least rank m. We call such a posture, θ, *locally fault tolerant*.

The Jacobian of a k-ROD, J_{ROD}, can be obtained by deleting the columns of the fault-free Jacobian that correspond to the frozen DOFs; the dimensions of J_{ROD} are $m \times (n-k)$. A necessary condition for J_{ROD} to be of rank m is that n has to be larger than or equal to $(m+k)$. Indeed, the manipulator needs to have at least m functional DOFs, even after a failure of k of them. That means that k DORs are necessary for local fault tolerance.

Are k DORs also *sufficient* for local fault tolerance? Consider a manipulator with $n = m+k$ DOFs; the

Jacobian of a k-ROD, J_{ROD}, is then a square $m \times m$ matrix. A posture, $\theta \in \Re^n$, is locally fault tolerant if the Jacobians of all m RODs are full rank. When the rank of any J_{ROD} is less than m, the robot is in an *internal singularity.* The difference between singular, locally fault tolerant and internally singular postures is illustrated in Figure 5. The locus of internal singularities is a set of $(m+k-1)$-dimensional surfaces in T^n; or $(n-1)$-dimensional surfaces, when $n = m+k$. Thus, nearly all postures of a manipulator with k DORs are locally k-fault tolerant. This can be summarized in the following theorem.

Theorem:

k degrees-of-redundancy are necessary and sufficient for k-th order local fault tolerance.

The fact that a posture is locally k-fault tolerant guarantees that the manipulator can reach every point in a neighborhood of the end effector position, even after failure of k DOFs. However, this neighborhood can be small. To extend this result to larger trajectories, we have to formulate a *global* fault tolerance condition. This can best be illustrated with an example.

4.2 Example of Task Specific Fault Tolerance

Before we present the global fault tolerance condition, we modify the definition of fault tolerance to include task specificity:

- **Task Specific Fault Tolerant Manipulator**: A manipulator is 1-fault tolerant with respect to the task of following the Cartesian path, $p(t)$, if and only if there exists a fault tolerant joint trajectory, $\theta(t)$.

- **Fault tolerant joint trajectory:** A trajectory, $\theta(t) \in \Re^n$, is 1-fault tolerant with respect to

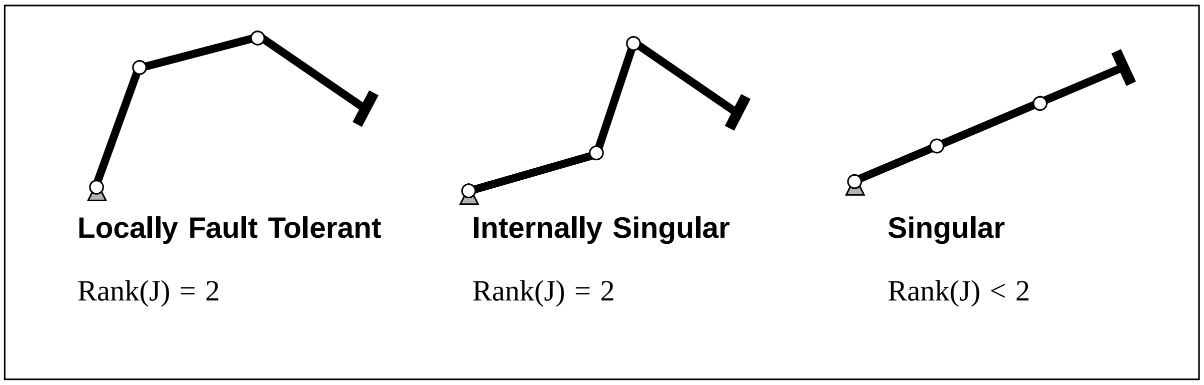

Figure 5: Examples of locally fault tolerant, internally singular and singular postures of a 3-DOF planar manipulator.

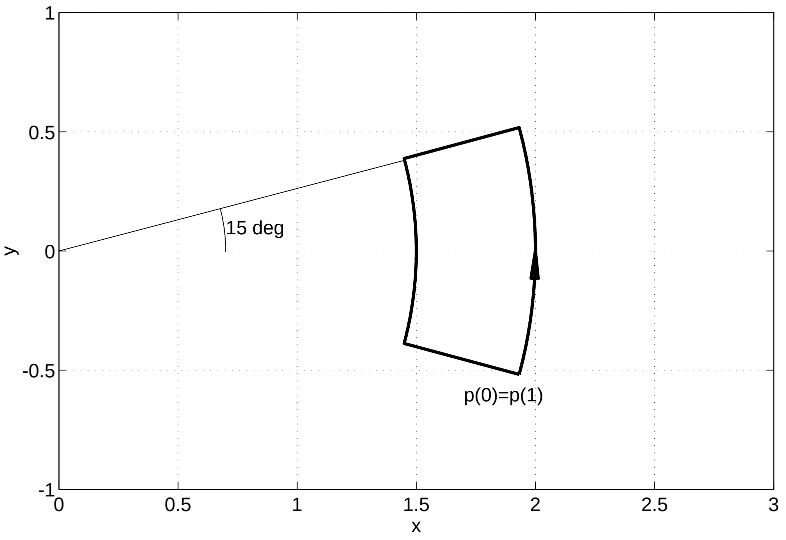

Figure 6: Path for the example of task specific fault tolerance.

the task of following the Cartesian path $p(t) \in \Re^m$, if for every DOF $j = 1 \ldots n$ and for every instant, ${}^f t$, there exists an alternate trajectory, $\theta(t, j, {}^f t)$, for which:

1) $\theta(t, j, {}^f t)$ maps onto $p(t)$ under the forward kinematics
2) $)(t) = \theta(t, j, {}^f t) \qquad \forall t \le {}^f$
3) $_j(t, j, {}^f t) = \theta_j({}^f t) \qquad \forall t > {}^f$ (i.e., the j-th DOF is frozen at the time ${}^f t$)

There are two main difference between this definition and the one for general purpose fault tolerance. The first difference is that we assume that the task is to follow a specific Cartesian path, rather than an unspecified path inside the task space. The second difference is that we no longer require that every point along the Cartesian path be reachable when a joint fails at an *arbitrary* angle, but only at an angle that occurred previously along the fault tolerant joint trajectory. Under these assumptions, k-fault tolerance can be achieved with only k redundant DOFs.

As an example, consider a 3-DOF planar manipulator with normalized link lengths of 1. We want to determine whether this manipulator is 1-fault tolerant with respect to the task of following the path, shown in Figure 6, which is parametrized as $p(\alpha)$ with $0 \le \alpha \le 1$. For every point, $p(\alpha)$, along the path, one can compute a preimage.[3] Because the manipulator in this example has one DOR, the preimage of every point $p(\alpha)$ is a one-dimensional subset of T^n, and can be parametrized in this case as

3. The preimage of a point p is the set of all postures, θ, for which $f(\theta) = p$, where f is the forward kinematics mapping of the manipulator.

$\theta = g(p(\alpha), \beta)$ with $\beta \in T^1$ (In general, the preimage of a point might consist of disjoint manifolds, in which case each manifold should be parametrized separately). The continuous function, g, describes a 2-dimensional surface in T^3, as is shown in Figure 7. Any joint trajectory that maps onto the specified Cartesian path, $p(\alpha)$, can be formulated as $\beta(\alpha)$, or $\theta(\alpha) = f(p(\alpha), \beta(\alpha))$. According to the definition of task specific fault tolerance, the manipulator is fault tolerant if and only if a fault tolerant trajectory, $\theta(\alpha)$, can be found. It is clear that every posture of a fault tolerant trajectory, $\theta(\alpha)$, has to be locally fault tolerant. However, this requirement is not sufficient because a fault at a point, $p(\alpha_1)$, might cause another point, $p(\alpha_2)$, to become unreachable, even if the posture $\theta(\alpha_1)$ were locally fault tolerant. Therefore, one should exclude as possible postures for a fault tolerant trajectory not only internally singular postures, but also postures that, in the case of failure, would cause an internal singularity further along the path. This is the condition for global fault tolerance. For our example, the set of acceptable postures is shown in Figure 8. The same set of postures can be represented in the (α, β)-plane—the white regions in Figure 9. A fault tolerant trajectory exists when a continuous function, $\beta(\alpha)$ with $0 \le \alpha \le 1$, can be found for which all postures, $\theta = f(p(\alpha), \beta(\alpha))$, are acceptable, i.e., satisfy the global fault tolerance condition. One such trajectory is shown in dashed line in Figure 9.

Further analysis of Figure 9 reveals that the unacceptable areas can be classified according to the failing DOF. This is illustrated in Figures 10, 11 and 12 for failures of DOFs 1, 2 and 3, respectively. The dotted

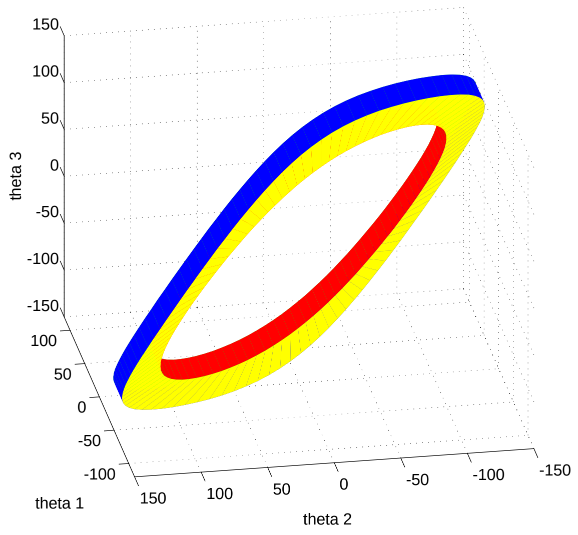

Figure 7: The preimage of a path.

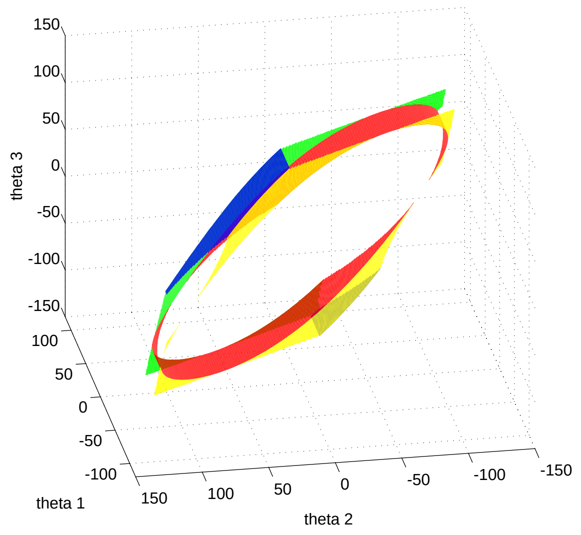

Figure 8: The set of acceptable points for a fault tolerant trajectory.

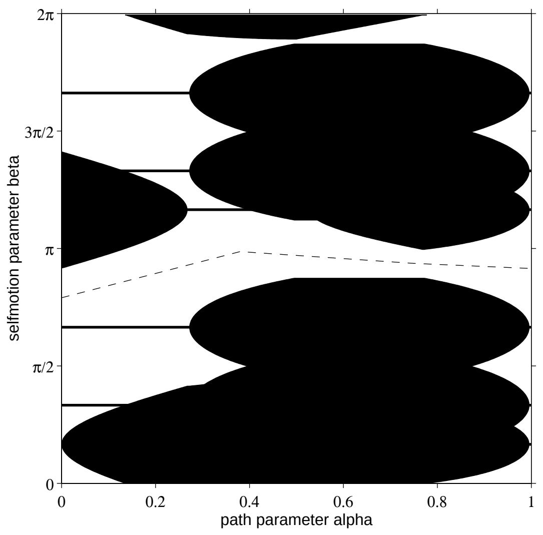

Figure 9: A possible fault tolerant trajectory. Regions of unacceptable postures, (α, β), are marked in gray.

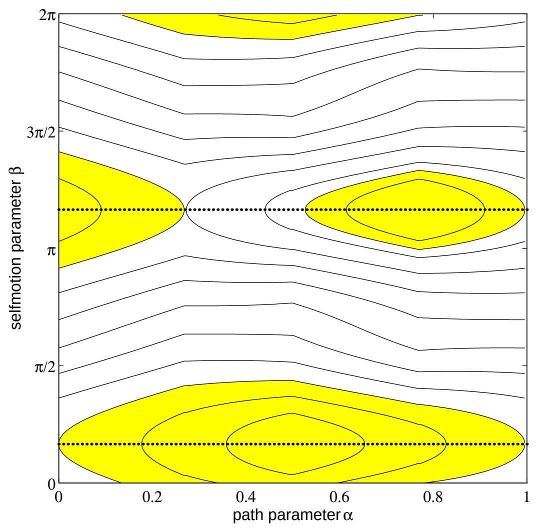

Figure 10: The unacceptable postures in case of a failure of joint 1 (areas in gray). The solid lines are contours of constant values of θ_1 and the dotted lines indicate internally singular postures.

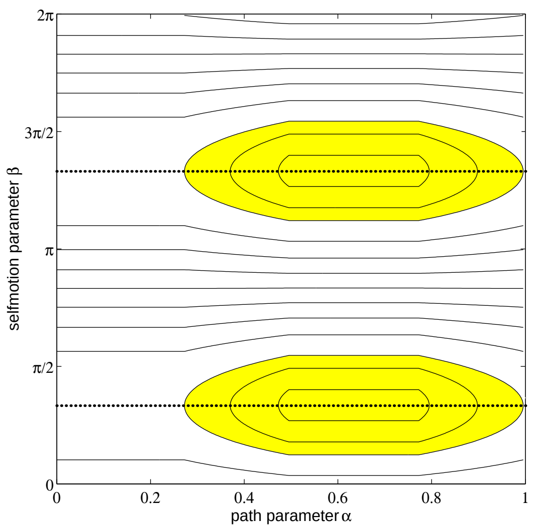

Figure 11: The unacceptable postures in case of a failure of joint 2.

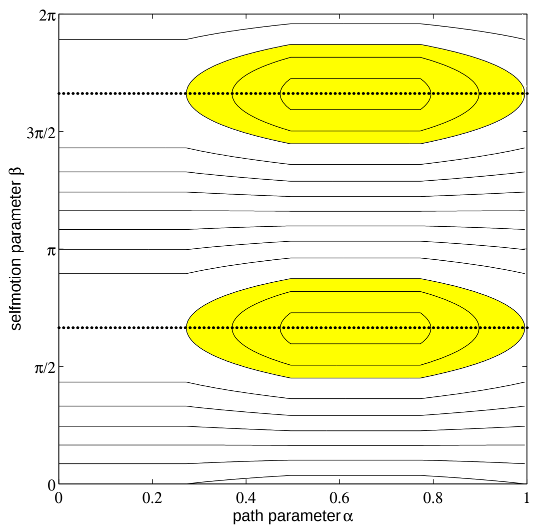

Figure 12: The unacceptable postures in case of a failure of joint 3.

lines correspond to internally singular postures, while the solid lines are contours of constant joint angles θ_1, θ_2 and θ_3, respectively. These contours represent the joint trajectories that are followed once a failing DOF has been immobilized. The gray areas are postures for which the global fault tolerance condition is violated. Notice that the contour lines in these areas do not reach the end of the Cartesian path, $\alpha = 1$, but instead get stuck at an internal singularity. It is also important to notice that the acceptable areas (in white) can be described by *fault tolerant joint intervals* for each of the DOFs. For example, the area in which the fault tolerant joint trajectory is drawn in Figure 9, can be characterized by the joint intervals depicted in Figure 13. As long as the joint angles stay within these fault tolerant joint intervals at each point along the path, fault tolerant execution of the path is guaranteed.

4.3 Redundancy Resolution Algorithms for Task Specific Fault Tolerance

We have shown that, to fault tolerantly follow a Cartesian path, it is important to follow a joint space trajectory that consists only of postures satisfying the global fault tolerance condition. How can we achieve this practically?

An obvious implementation is to compute a fault tolerant joint trajectory off-line and store it as a set of via points in joint space. At execution time, a joint space controller can be used to follow this trajectory until a joint failure is detected. At that point, the failing joint is immobilized and an inverse kinematics algorithm is used on-line to compute the appropriate joint angles for the remaining DOFs. Although

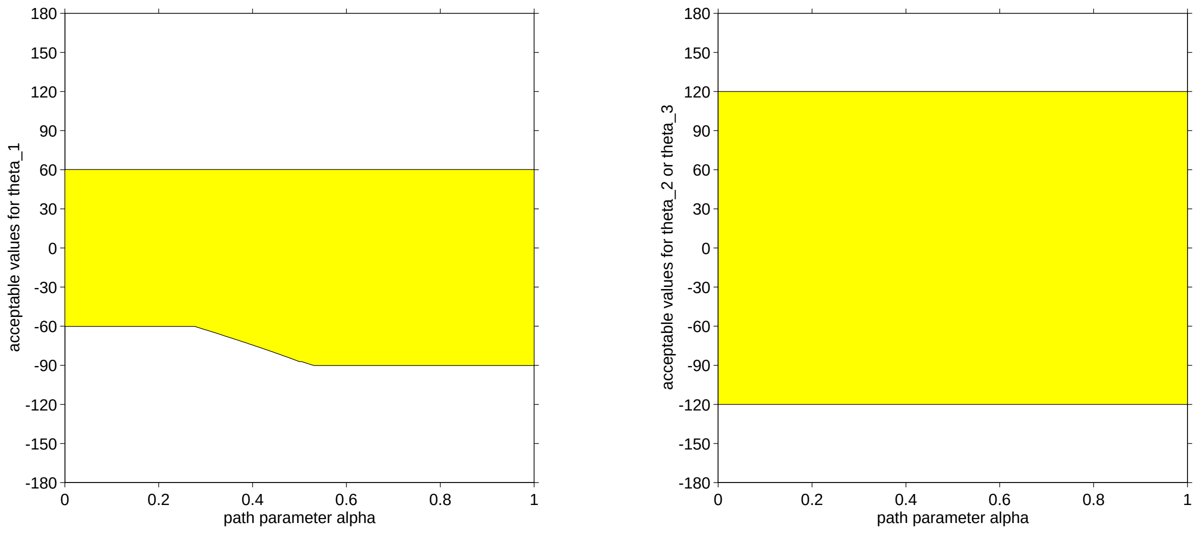

Figure 13: Fault tolerant joint intervals for the trajectory shown in dashed line in Figure 9.

simple and computationally inexpensive, this scheme has several disadvantages. First, the fault tolerant joint trajectory cannot be adjusted at run time to satisfy secondary task requirements, such as obstacle avoidance. Also, when a failure occurs, one has to change the joint trajectory generation algorithm instantaneously, making it difficult to achieve a smooth transition from operation with n DOFs to operation with $(n-1)$ DOFs.

To avoid these drawbacks, we suggest that one uses a kinematic controller of the form[4]:

$$\dot{\theta} = J^{\dagger}\dot{x} + (I - J^{\dagger}J)\nabla h, \tag{14}$$

both before and after a failure has occurred, as is also suggested in Lewis and Maciejewski (1994b). The transition to operation with one fewer DOF can simply be achieved by zeroing out the column of the Jacobian that corresponds to the immobilized DOF. The gradient projection method (Liégeois 1977) can be used to assure that the manipulator stays within the fault tolerant joint intervals, indicated in Figure 13. By choosing an appropriate potential function, h, one can even combine the requirement for fault tolerance with a secondary task requirement. An additional advantage is that this scheme can accommodate for small changes in the Cartesian path, while still maintaining the property of fault tolerance. As long as the joint trajectory stays within the fault tolerant joint intervals, it is guaranteed that every point further along the Cartesian path will be reachable even after joint failure. This is particularly important in sensor based systems in which small trajectory corrections are made based on sensor feedback.

5 Summary

In this article, we have shown that making a manipulator fault tolerant by adding redundant DOFs is an effective way to increase the reliability of a manipulator. However, not every redundant manipulator is fault tolerant. Thus, an important problem for the design of fault tolerant manipulators is: How many DOFs are necessary and sufficient for fault tolerance and how should these DOFs be distributed along the length of the manipulator? We have shown that, depending on the assumptions that are made about the task, the answer to this question varies.

For general purpose fault tolerant manipulators without joint limits, two degrees-of-redundancy are necessary and sufficient to sustain one fault. This conclusion was illustrated with two spatial general

4. Readers who are unfamiliar with local optimization of kinematic redundancy are referred to Nakamura (1991) for a detailed tutorial.

purpose fault tolerant manipulator designs: a 5-DOF positional manipulator and an 8-DOF positional and orientational manipulator. Both manipulators have a fault tolerant workspace without any holes or voids so that one can scale the designs to fit any task.

For task specific fault tolerant manipulators, only one degree-of-redundancy is necessary and sufficient for 1-fault tolerance. However, one might have to use a different manipulator and recompute the fault tolerant joint intervals, for every task. This drawback can be partially overcome by using a modular manipulator system that can be quickly reconfigured to suit a particular task.

Acknowledgments

This research was funded in part by the Department of Energy under grant DOE-F902-89ER14042, by Sandia National Laboratories under contract AC-3752-A, by the Department of Electrical and Computer Engineering, and by The Robotics Institute at Carnegie Mellon University.

The authors would also like to thank the reviewers for pointing out some flaws in the original manuscript. Thanks to their comments and suggestions the quality of this article has been improved considerably.

References

Burdick, J. W. 1988. Kinematic analysis and design of redundant robot manipulators. STAN-CS-88-1207. Stanford, Calif.: Stanford University Computer Science.

Burdick, J. W. 1992 (May 12–14, Nice, France). A recursive method for finding revolute-jointed manipulator singularities. *Proc. 1992 IEEE Int. Conf. Robot. Automat.* Los Alamitos, Calif.: IEEE, pp. 448–453.

Chen, I.-M., and Burdick, J. W. 1995 (May 21-27, Nagoya). Determining task optimal modular robot assembly configurations. *Proc. 1995 IEEE Int. Conf. Robot. Autom.* Los Alamitos, Calif.: IEEE.

Chow, E. Y., and Willsky, A. S. 1984. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. Automat. Contr.* 29(7):603–614.

Fukuda, T., et al. 1992. Concept of cellular robotic system (CEBOT) and basic strategies for its realization. *Comput. Electr. Eng.* 18(1):11–39.

Hui, R., et al. 1993 (May 2–6, Atlanta). Design of the IRIS facility—a modular, reconfigurable and expandable robot test bed. *Proc. 1993 IEEE Int. Conf. Robot. Autom.* Los Alamitos, Calif.: IEEE, pp. 155–160.

Johnson, B.W. 1989. *Design and analysis of fault-tolerant digital systems*. Reading, Mass.: Addison-Wesley.

Lewis, C. L., and Maciejewski, A. A. 1994a. Dexterity optimization of kinematically redundant manipulators in the presence of joint failures. *Comput. Electr. Eng.* 20(3):273–288.

Lewis, C. L., and Maciejewski, A. A. 1994b (May 8–13, San Diego). An example of failure tolerant operation of a kinematically redundant manipulator. *Proc. 1994 IEEE Int. Conf. Robot. Autom.* Los Alamitos, Calif.: IEEE, pp. 1380–1387.

Liégeois, A. 1977. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. Syst. Man Cybern.* 7:868–871.

Nakamura, Y. 1991. *Advanced robotics: redundancy and optimization.* Reading, Mass.: Addison-Wesley.

Paredis, C. J. J., and Khosla, P. K. 1993. Kinematic design of serial link manipulators from task specifications. *Int. J. Robotics Res.* 12(3):274–287.

Paredis, C. J. J., and Khosla, P. K. 1994 (May 8–13, San Diego). Mapping tasks into fault tolerant manipulators. *Proc. 1994 IEEE Int. Conf. Robot. Autom.* Los Alamitos, Calif.: IEEE, 696–703.

Pradeep, A. K., et al. 1988. Crippled motion in robots. *IEEE Trans. Aerosp. Electron. Syst.* 24(1):2–13.

Schmitz, D. E., Khosla, P. K., and Kanade, T. 1988 (Nov. 6–10, Sydney). The CMU reconfigurable modular manipulator system. *Proc. 19th Int. Symp. Exp. Rob. (ISIR)*. New York: Springer-Verlag, pp. 473–488.

Sreevijayan, D. 1992. On the design of fault-tolerant robotic manipulator systems. M.S. thesis. University of Texas at Austin Department of Mechanical Engineering.

Stengel, R. F. 1988. Intelligent failure-tolerant control. *IEEE Control Syst. Mag.* 11(4):2–13.

Ting, Y., Tosunoglu, S., and Tesar, D. 1993 (May 2–6, Atlanta). A control structure for fault-tolerant operation of robotic manipulators. *Proc. 1993 IEEE Int. Conf. Robot. Autom.* Loa Alamitos, Calif.: IEEE, pp. 684–690.

Visinsky, M. L., Walker, I. D., and Cavallaro, J. R. 1993 (May 2–6, Atlanta). Layered dynamic fault detection and tolerance for robots. *Proc. 1993 IEEE Int. Conf. Robot. Autom.* Los Alamitos, Calif: IEEE, pp. 180–187.

Visinsky, M. L., Walker, I. D., and Cavallaro, J. R. 1994 (May 8–13, San Diego). New dynamic model-based fault detection thresholds for robot manipulators. *Proc. 1994 IEEE Int. Conf. Robot. Autom.* Los Alamitos, Calif.: IEEE, pp. 1388–1395.

von Neumann, J. 1956. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata studies (Annals of mathematics studies, no. 34),* eds. C. E. Shannon and J. McCarthy. Princeton: Princeton University Press.

Wu, E. C., Hwang, J. C., and Chladek, J. T. 1993. Fault-tolerant joint development for the Space Shuttle remote manipulator system: analysis and experiment. *IEEE Trans. Robot. Autom.* 9(5):675–684.