Real-Time Failure-Tolerant Control of Kinematically Redundant Manipulators

K. N. Groom, A. A. Maciejewski and V. Balakrishnan

Purdue University
1285 Electrical Engineering Building
West Lafayette, Indiana 47907-1285

ABSTRACT

This work considers real-time fault-tolerant control of kinematically redundant manipulators to single locked-joint failures. The fault-tolerance measure used is a worst-case quantity, given by the minimum, over all single joint failures, of the minimum singular value of the post-failure Jacobians. Given any end-effector trajectory, the goal is to continuously follow this trajectory with the manipulator in configurations that maximize the fault-tolerance measure. The computation required to track these optimal configurations with brute-force methods is prohibitive for real-time implementation. We address this issue by presenting algorithms that quickly compute estimates of the worst-case fault-tolerance measure and its gradient. Real-time implementations are presented for all these techniques, and comparisons show that the performance of the best is indistinguishable from that of brute-force implementations.

I. INTRODUCTION

Robot failures are not uncommon. A report from the Japanese Ministry of Labor indicates that over 60% of the industrial robots studied had a mean-time-between-failure of less than 500 hours; indeed, 28.7% had mean-time-between-failure of 100 hours or less [1]. Similar numbers can be derived for robots in nonindustrial environments, by deducing mean-time-between-failure from reliability data for individual components. Table I gives the mean-time-to-failure and reliability $R(x)$ (probability that the component is still functioning after x hours) for typical robot components and subsystems employed in a ground mobile environment [2]. Assuming that the components fail independently of each other, it can be shown that eight out of ten robots will likely fail after 1000 hours of operation (for six DOF

TABLE I

Device	MTTF	$R(1000)$	$R(10,000)$
Servo Amplifier	136,054	0.993	0.929
DC Servo Motor	31,519	0.969	0.728
Gear Box	53,319	0.981	0.829
Optical Encoder	4,845	0.814	0.127
Tachometer	9,606	0.901	0.353

robots, with each joint consisting of a servo amplifier, servo motor, gear box and optical encoder). More severe environmental conditions will, of course, result in even worse failure rates.

Failures in robots have significant consequences, ranging from economic impact in industrial applications

to potentially catastrophic incidents in remote and/or hazardous environments. A direct approach towards increasing robot reliability is to improve the reliability of the individual components; however, achieving acceptable reliability rates is often prohibitively expensive, and sometimes technologically impossible. An alternative approach is to consider failure-tolerant robot designs. These typically incorporate a failure detection and identification scheme [3] followed by failure recovery [4]. Designing the robot with redundant systems increases the options available for failure tolerance. Redundancy can be in the form of duplication of actuators and sensors [5, 6], or in the form of kinematic redundancy [7, 8, 9]. Proper utilization of kinematic redundancy provides greater dexterity prior to failures, minimizes the immediate impact of a failure, and guarantees task completion by ensuring a reachable post-failure workspace.

This work presents real-time implementation of schemes for utilizing kinematic redundancy to maximize the tolerance of robots to single locked-joint failures. The fault-tolerance measure used is a worst-case quantity, given by the minimum, over all single joint failures, of the minimum singular value of the post-failure Jacobians. Maximizing this measure corresponds to configuring the robot to minimize the worst-case end-effector velocity error over all single locked-joint failures. This is also equivalent to minimizing the worst-case discontinuities in the joint velocities of the healthy joints as they move to compensate for the failed joint. The main contributions of this work are computationally efficient schemes for computing the fault-tolerance measure and its gradient, which in turn enable real-time optimal configuration of robots in anticipation of failures.

II. FRAMEWORK FOR FAILURE-TOLERANT CONTROL

The forward kinematics of manipulators are frequently represented as

$$\dot{x} = J\dot{q}. \qquad (1)$$

where $\dot{x}$ is an m-dimensional vector representing the end-effector velocity, $\dot{q}$ is an n-dimensional vector describing the joint velocities, and J is the m by n manipulator Jacobian matrix [10]. For a redundant manipulator, $n > m$ so the equation is underconstrained and there are an infinite number of solutions which can be expressed as

$$\dot{q} = J^+\dot{x} + \left(I - J^+J\right)z, \qquad (2)$$

where J^+ is the pseudoinverse of the Jacobian, and $\mathbf{z} \in \mathbb{R}^n$. The first term on the right in (2) corresponds to the least-squares minimum-norm solution, while the second is the projection of the vector $\mathbf{z}$ onto the null space of the Jacobian. The vector $\mathbf{z}$ is frequently chosen as $\mathbf{z} = \nabla h(\mathbf{q})$ in order to optimize h, under exact end-effector tracking [11]. Other methods for optimizing h, such as those based on the extended Jacobian [12] or the augmented Jacobian [13, 14], also require its gradient.

The Singular Value Decomposition (SVD) provides a mathematical framework for describing both the optimization scheme (2), as well as the failure-tolerance measure considered here. The SVD of the Jacobian is the matrix factorization

$$J = U\Sigma V^T \qquad (3)$$

where $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix of the left singular vectors $\hat{\mathbf{u}}_i$, and $V \in \mathbb{R}^{n \times n}$ is an orthogonal matrix of the right singular vectors $\hat{\mathbf{v}}_i$. The matrix Σ is m by n, with $\Sigma = [\Sigma_d \; \mathbf{0}]$, where $\Sigma_d = \mathrm{diag}(\sigma_1, \ldots, \sigma_m)$ and $\mathbf{0}$ is an m by $n - m$ zero matrix. The σ_i are called the singular values of J, and satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. The rank r of J is simply the number of its nonzero singular values. The SVD can be used to compute the pseudoinverse of the Jacobian

$$J^+ = V\Sigma^+ U^T \qquad (4)$$

as well as the null space projection

$$\left(I - J^+ J\right) = \sum_{i=r+1}^{n} \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^T. \qquad (5)$$

The SVD has long been a valuable tool for quantifying various dexterity measures: manipulability [15] (product of the singular values), isotropy [16] (ratio of the maximum singular value to the minimum), task compatibility [17] (weighted combination of singular values) and proximity to singularities [18] (minimum singular value). Each of these measures has its own physical interpretation. In particular, the minimum singular value, besides defining the distance from singularity, also has the property that its reciprocal gives the worst-case joint velocity norm over all desired unit-norm end-effector velocities. Worst-case measures are arguably the most appropriate for fault-tolerance studies, since guarantees of a certain level of performance are required. These observations motivate the following definition of the kinematic fault-tolerance measure:

$$\mathcal{K} = \min_{f=1\ldots n} \sigma_m \left({}^f J\right), \qquad (6)$$

where

$${}^f J = \begin{bmatrix} \mathbf{j}_1 & \mathbf{j}_2 & \cdots & \mathbf{j}_{f-1} & \mathbf{0} & \mathbf{j}_{f+1} & \cdots & \mathbf{j}_n \end{bmatrix} \qquad (7)$$

is the Jacobian following the locked-joint failure of the fth joint, obtained by simply zeroing out the fth column of the original Jacobian [8]. We also define the index of the most debilitating joint failure as

$$F = \arg \min_{f=1\ldots n} \sigma_m \left({}^f J\right). \qquad (8)$$

The gradient of the function $\mathcal{K}$ can be computed and used in the gradient projection scheme suggested by (2), thereby maximizing the manipulator's failure tolerance while exactly tracking $\dot{\mathbf{x}}$ [19]. The kinematic fault-tolerance measure can be re-expressed in terms of the singular vectors of the failed Jacobian ${}^F J$ as

$$\mathcal{K} = {}^F\hat{\mathbf{u}}_m^T \; {}^F J \; {}^F\hat{\mathbf{v}}_m. \qquad (9)$$

The change of $\mathcal{K}$ with respect to q_i is then given by

$$\frac{\partial \mathcal{K}}{\partial q_i} = {}^F\hat{\mathbf{u}}_m^T \; \frac{\partial {}^F J}{\partial q_i} \; {}^F\hat{\mathbf{v}}_m. \qquad (10)$$

For rotary-jointed robots [1], the partial derivative of the kth column of the Jacobian with respect to q_i can be efficiently computed as follows [20]:

$$\partial \mathbf{j}_k / \partial q_i = \begin{cases} \begin{bmatrix} (\hat{\mathbf{z}}_i^T \mathbf{p}_k)\hat{\mathbf{z}}_k - (\hat{\mathbf{z}}_i^T \hat{\mathbf{z}}_k)\mathbf{p}_k \\ \hat{\mathbf{z}}_i \times \hat{\mathbf{z}}_k \end{bmatrix}, & i < k \\[2ex] \begin{bmatrix} (\hat{\mathbf{z}}_k^T \mathbf{p}_i)\hat{\mathbf{z}}_i - (\hat{\mathbf{z}}_i^T \hat{\mathbf{z}}_k)\mathbf{p}_i \\ \mathbf{0} \end{bmatrix}, & i \geq k \end{cases} \qquad (11)$$

where $\hat{\mathbf{z}}_l$ is the axis of rotation of the lth joint and $\mathbf{p}_l$ is the vector from the lth joint axis to the end effector. The gradient of $\mathcal{K}$ can then be computed from (10) and (11):

$$\nabla \mathcal{K} = \begin{bmatrix} \dfrac{\partial \mathcal{K}}{\partial q_1} & \dfrac{\partial \mathcal{K}}{\partial q_2} & \cdots & \dfrac{\partial \mathcal{K}}{\partial q_n} \end{bmatrix}^T. \qquad (12)$$

The various steps in a numerical implementation of the the proposed failure-tolerant control algorithm are:

1. Compute the pseudoinverse and the null space projection of the Jacobian.

2. Compute the kinematic fault-tolerance measure $\mathcal{K}$ and the associated failed column F of the Jacobian.

3. Compute the gradient of the kinematic fault-tolerance measure $\nabla \mathcal{K}$.

The most computationally intensive step in the algorithm is the second step, since it requires the computation of the minimum singular value for the n possible joint failures. This makes a naive implementation of (6) unsuitable for real-time applications. In addition, as with any gradient optimization technique applied to this measure, the most debilitating joint failure will typically be nonunique. From a practical standpoint, this leads to "chattering" (i.e., switching back and forth between multiple worst-case joint failures). This can be addressed by keeping track of all "near worst-case" joints, thus exacerbating the computational cost. Therefore the issue of real-time implementation is a significant one; this will be the focus of the remainder of the paper.

---

[1] Prismatic joints simplify the expressions.

III. REAL-TIME CALCULATION OF $\mathcal{K}$ AND $\nabla\mathcal{K}$

Fundamental bounds on the change in the singular values of the Jacobian following a locked-joint failure can be derived. In particular, the decrease in any singular value of a matrix following the removal of a column is bounded by the norm of that column. Results such as these can in turn be used to derive useful lower bounds for $\mathcal{K}$, and possibly eliminate candidate Fs.

We present three techniques for real-time computation of $\mathcal{K}$ and its gradient. The first two use a Taylor series approximation to extrapolate the singular values and singular vectors from their pre-failure values. The third approach relies on the fact that the changes in the joint variables along any trajectory are continuous, and therefore for small increments along a trajectory, power methods can be used for efficient tracking.

For all three techniques presented here, it is assumed that the full SVD of the pre-failure Jacobian is available, as a result of completing Step 1.

A. First Order Polynomial Approximation

Let

$$^fJ(\alpha) = [\, \mathbf{j}_1 \cdots \mathbf{j}_{f-1} \ (1-\alpha)\mathbf{j}_f \ \mathbf{j}_{f+1} \cdots \mathbf{j}_n \,]. \quad (13)$$

The case when $\alpha = 0$ corresponds to a healthy manipulator, while $\alpha = 1$ corresponds to a locked-joint failure of the fth joint. A linear approximation of the the singular values for $^fJ(1)$ can be obtained from the SVD of $J(0)$ as

$$^f\sigma_i(1) = {}^f\sigma_i(0) + \frac{d\,^f\sigma_i}{d\alpha}\bigg|_{\alpha=0}, \quad (14)$$

where

$$\frac{d\,^f\sigma_i}{d\alpha} = {}^f\hat{\mathbf{u}}_i^T \frac{d\,^fJ}{d\alpha} {}^f\hat{\mathbf{v}}_i, \quad (15)$$

and

$$\frac{d\,^fJ}{d\alpha} = [\, \mathbf{0}_1 \cdots \mathbf{0}_{f-1} \ -\mathbf{j}_f \ \mathbf{0}_{f+1} \cdots \mathbf{0}_n \,]. \quad (16)$$

Substituting (16) and (15) into (14) yields

$$^f\sigma_i(1) = \sigma_i(0) - \hat{\mathbf{u}}_i^T(0)\mathbf{j}_f\hat{\mathbf{v}}_{fi}(0). \quad (17)$$

Since $\hat{\mathbf{u}}_i(0)$ and $\hat{\mathbf{v}}_i(0)$ are available, the estimate of each $^f\sigma_i(1)$ requires only $2m + 1$ flops[2]. Similar estimates can be derived for $^f\hat{\mathbf{u}}_i(1)$ and $^f\hat{\mathbf{v}}_i(1)$, but we have found them to be quite poor. Therefore, we use (17) only to determine the worst-case joint failure index F; this requires using (17) to obtain estimates of all the singular values for all possible single locked-joint failures, and picking the failure that is the worst case. Note that for every joint failure, estimates of all the singular values need to be computed, since the ordering of the singular values may change. Once the worst-case joint-failure index F is estimated, we compute the minimum singular value and the associated singular vectors of ^FJ, required to calculate the gradient from (10).

---

[2]A flop is defined as one floating point operation (add or multiply).

The accuracy of the first-order approximation of even the change in the singular values may be quite poor. For the example illustrated in Fig. 1, a strong interaction between the singular values $^f\sigma_3$ and $^f\sigma_4$ can be seen. This is largely due to the fact that they are
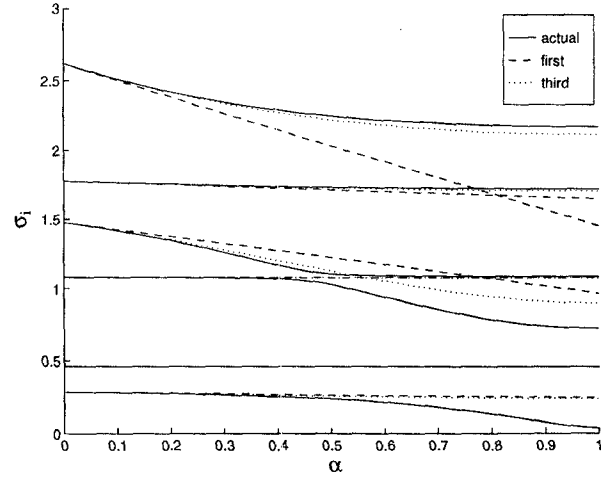

Fig. 1. An example of the change in singular values of a typical $J_{6\times7}$ as a column is removed.

"close" to each other. The singular values $^f\sigma_1$ and $^f\sigma_6$ show considerable interaction as well. It turns out that in this example, the failed joint contributes considerably to the singular value $^f\sigma_1$, and therefore the local linear estimate of this singular value predicts a large drop. However, other joints also contribute to this singular value; in particular, as the column associated with the failed joint is zeroed out, the singular vectors associated with $^f\sigma_1$ and $^f\sigma_6$ rotate, "transferring" the effect of the failed joint from one to the other. This rate of change of the singular vectors is not considered in (17); this motivates retaining more terms in the Taylor series expansion of $^f\sigma_i(\alpha)$. We explore this in the next section.

B. Third Order Polynomial Approximation

We have already seen how the interaction between the singular values can be intuitively explained in terms of rotation of the associated singular vectors. These comments can be made more precise with an explicit derivation of the rate of change in the singular vectors:

$$\frac{d\hat{\mathbf{u}}_i}{d\alpha} = 2(1-\alpha)\hat{\mathbf{u}}_i^T \mathbf{j}_f \sum_{\substack{k=1 \\ k\neq i}}^{m} \frac{\hat{\mathbf{u}}_k^T \mathbf{j}_f}{\sigma_k^2 - \sigma_i^2} \hat{\mathbf{u}}_k \quad (18)$$

$$\frac{d\hat{\mathbf{v}}_i}{d\alpha} = (1-\alpha)\hat{\mathbf{u}}_i^T \mathbf{j}_f \sum_{\substack{k=1 \\ k\neq i}}^{m} \frac{\hat{\mathbf{u}}_k^T \mathbf{j}_f \left(\frac{\sigma_i}{\sigma_k} + \frac{\sigma_k}{\sigma_i}\right)}{\sigma_k^2 - \sigma_i^2} \hat{\mathbf{v}}_k$$

$$- \hat{\mathbf{u}}_i^T \mathbf{j}_f \sum_{k=m+1}^{n} \frac{v_{fk}}{\sigma_i} \hat{\mathbf{v}}_k. \quad (19)$$

Two of the major factors that we discussed as influencing the rate of change of the singular vectors associated with a particular singular value can now be clearly seen: how close other singular values are to this singular value (the $1/(\sigma_k^2 - \sigma_i^2)$ factors), and how much the failed joint contributes to that singular value (the $\mathbf{u}_i^T \mathbf{j}_f$ term).

This additional information about the rate of change of singular vectors can be used to derive higher order estimates for $^f\sigma_i(\alpha)$. From (15) (and noting that $d^2\,^fJ/d\alpha^2 = 0$), we have

$$\frac{d^2\,^f\sigma_i}{d\alpha^2} = \frac{d\,^f\hat{\mathbf{u}}_i^T}{d\alpha}\frac{d\,^fJ}{d\alpha}{}^f\hat{\mathbf{v}}_i + {}^f\hat{\mathbf{u}}_i^T\frac{d\,^fJ}{d\alpha}\frac{d\,^f\hat{\mathbf{v}}_i}{d\alpha}. \qquad (20)$$

It can be shown, using standard arguments from linear algebra that

$$\left.\frac{d\,^f\sigma_i}{d\alpha}(\alpha)\right|_{\alpha=1} = 0. \qquad (21)$$

Using the four boundary conditions

1. $\sigma_i(0)$, given by the pre-failure SVD;
2. $d^f\sigma_i/d\alpha$ at $\alpha = 0$, given by (15);
3. $d^f\sigma_i/d\alpha$ at $\alpha = 1$, given by (21);
4. $d^2\,^f\sigma_i/d\alpha^2$ at $\alpha = 0$, given by (20),

we derive a third-order approximation for $^f\sigma_i(\alpha)$:

$$\begin{aligned}^f\sigma_i(\alpha) &= \sigma_i(0) + {}^f\sigma_i'(0)\alpha + \tfrac{1}{2}{}^f\sigma_i''(0)\alpha^2 \\ &\quad - \tfrac{1}{3}\left({}^f\sigma_i'(0) + {}^f\sigma_i''(0)\right)\alpha^3,\end{aligned} \qquad (22)$$

where "$'$" denotes differentiation with respect to α.

The behavior of the third order polynomial approximation as compared to the first order is illustrated in Fig. 1. The third-order approximations are in general more accurate, particularly for large singular values as in the case of σ_1, however, large discrepancies in their estimates of small singular values are not uncommon. This is once again due to the complex interaction between singular values that have strong contributions from the failed joint. The fundamental problem is that the failure of a column can be a significant perturbation, and can require a large number of terms in the Taylor series expansion. This motivates the exploration of an alternate technique.

C. Power Method Approximation

Since the underlying objective is one of tracking the singular values and vectors of manipulator Jacobians, which varies smoothly with the joint variables $\mathbf{q}$, it is natural to consider power methods. Power methods have been successfully applied to track the minimum singular value and the associated singular vectors for the purpose of traversing singular configurations [21]. These iterative methods converge very rapidly—typically in one or two iterations—if a good initial estimate of the SVD is available (which is the case here). However, each iteration of the power method is usually quite expensive. It will be shown that for our application, a computationally efficient scheme can be devised, making real-time implementation feasible.

In order to compute $\mathcal{K}$ and its gradient, we need to track the index of the failed joint F, as well as the minimum singular value and one of the associated singular vectors of ^FJ. In general, F will change along a trajectory, and therefore, we need to track the minimum singular values and one of the associated singular vectors of all n post-failure Jacobians.

To estimate the current minimum singular value $^f\sigma_m$ and the associated left singular vector $^f\hat{\mathbf{u}}_m$, we need to apply the power method to the matrix $\left(^fJ\,^fJ^T\right)^{-1}$. We write

$$^fJ\,^fJ^T = J\,J^T - \mathbf{j}_f\mathbf{j}_f^T, \qquad (23)$$

which can be re-expressed in terms of the SVD of J as

$$\left(^fJ\,^fJ^T\right) = U\left(\Sigma\Sigma^T - U^T\mathbf{j}_f\mathbf{j}_f^T U\right)U^T. \qquad (24)$$

Therefore,

$$U^T\left(^fJ\,^fJ^T\right)^{-1}U = \left(\Sigma\Sigma^T - U^T\mathbf{j}_f\mathbf{j}_f^T U\right)^{-1}. \qquad (25)$$

The matrices $\left(^fJ\,^fJ^T\right)^{-1}$ and $\left(\Sigma\Sigma^T - U^T\mathbf{j}_f\mathbf{j}_f^T U\right)^{-1}$ have the same singular values, and their singular vectors are related by the coordinate transformation

$$^f\hat{\mathbf{u}}_i = U\,^f\hat{\mathbf{w}}_i, \qquad (26)$$

where $^f\hat{\mathbf{w}}_i$ are the singular vectors of the latter matrix.

It turns out that applying the power method to the matrix on the right-hand side of (25) is significantly more efficient. Using the identity

$$\left(A + \mathbf{a}\mathbf{b}^T\right)^{-1} = A^{-1} - \frac{\left(A^{-1}\mathbf{a}\right)\left(\mathbf{b}^T A^{-1}\right)}{1 + \mathbf{b}^T A^{-1}\mathbf{a}} \qquad (27)$$

we get

$$\left(\Sigma\Sigma^T - U^T\mathbf{j}_f\mathbf{j}_f^T U\right)^{-1} = \qquad (28)$$

$$\left(\Sigma\Sigma^T\right)^{-1} + \frac{\left(\left(\Sigma\Sigma^T\right)^{-1}U^T\mathbf{j}_f\right)\left(\mathbf{j}_f^T U(\Sigma\Sigma^T)^{-1}\right)}{1 - \mathbf{j}_f^T U\left(\Sigma\Sigma^T\right)^{-1}U^T\mathbf{j}_f}.$$

We have assumed that $\Sigma\Sigma^T$ is nonsingular[3]; otherwise, we know that $\mathcal{K} = 0$. Noting that

$$\mathbf{j}_f^T U = \begin{bmatrix}\sigma_1 v_{f1} & \sigma_2 v_{f2} & \cdots \sigma_m v_{fm}\end{bmatrix}, \qquad (29)$$

we get

$$\left(\Sigma\Sigma^T - U^T\mathbf{j}_f\mathbf{j}_f^T U\right)^{-1} = \operatorname{diag}\left(1/\sigma_1^2, \ldots, 1/\sigma_m^2\right) + \qquad (30)$$

$$\frac{1}{\sum_{i=m+1}^n v_{fi}^2}\begin{bmatrix}v_{f1}/\sigma_1 \\ \vdots \\ v_{fm}/\sigma_m\end{bmatrix}\begin{bmatrix}v_{f1}/\sigma_1 & \cdots & v_{fm}/\sigma_m\end{bmatrix}.$$

---

[3]This case implies that the manipulator is already at a kinematic singularity; under our control scheme, this will occur only at a workspace boundary where there is no nonsingular configuration that achieves this task point.

The fundamental power iteration is then given by the following pseudo-code:

$$^f\mathbf{w}_m \quad := \quad \left(U^T \left(^f J \, ^f J^T\right)^{-1} U \right) {}^f\hat{\mathbf{w}}_m;$$

$$^f\sigma_m \quad := \quad \|^f\mathbf{w}_m\|;$$

$$^f\hat{\mathbf{w}}_m \quad := \quad {}^f\mathbf{w}_m/^f\sigma_m;$$

The matrix-vector product in the first step is performed using (30) and therefore requires at most $5m + 2n$ flops (while direct implementations of the inverse power method on $\left(^f J \, ^f J^T\right)^{-1}$ are considerably more expensive).

At the start of the power iteration, $^f\hat{\mathbf{w}}_m$ is initialized to its value from the end of the previous power iteration (performed for the configuration at the previous cycle time). Note that $^f\hat{\mathbf{w}}_m$ is the only quantity that has to be carried along from one cycle to another. At each cycle time, once the n power iterations are completed (one for each failure), $\mathcal{K}$, and its corresponding F can be immediately calculated. The associate left singular vector $^F\hat{\mathbf{u}}_m$ can be computed using (26); the right singular vector can be computed using

$$^F\hat{\mathbf{v}}_m = \frac{1}{^F\sigma_m} \left(^F\hat{\mathbf{u}}_m^T \, ^F J\right)^T. \tag{31}$$

Using these quantities and (10), the gradient of $\mathcal{K}$ can be computed.

IV. PERFORMANCE COMPARISON OF TECHNIQUES

The three techniques proposed for real-time implementation of optimizing failure tolerance were first tested on $10,000$ Jacobians of size 6×7, randomly distributed throughout the range of physically realizable rotary-jointed robots. Each column of these Jacobians is of the form

$$\mathbf{j}_i = \begin{bmatrix} \mathbf{v}_i \\ \hat{\omega}_i \end{bmatrix}$$

where $\hat{\omega}_i$ is of unit length, and pointing uniformly over all directions. Given $\hat{\omega}_i$, the vectors $\mathbf{v}_i$ have directions uniformly distributed over the subspace orthogonal to $\hat{\omega}_i$, with a length that is uniformly distributed over $[0, 2]$. This distribution is intended to represent a reasonably normalized Jacobian that has accounted for the disparity in units between linear and rotational velocities.

Since the power method uses information from the Jacobian of the previous cycle time, a single Jacobian does not provide enough information for testing this technique. To address this issue we generate a plausible configuration for the cycle time previous to that of each randomly generated Jacobian. This is done by first converting the random Jacobian to the corresponding manipulator defined by Denavit and Hartenberg parameters [22]. The previous configuration for this manipulator is then determined by applying a perturbation of 0.01 radians to each of the joints[4]. Finally, the SVD of

the Jacobian for this configuration is computed to obtain the previous $^f\hat{\mathbf{w}}_m$ required by the power method.

Figure 2 compares the accuracy of the three approximation methods for estimating the minimum singular value for each of the seven possible joint failures on the 10,000 test Jacobians. An analysis of this data can be summarized by noting that the performance of the power method was vastly superior to either of the polynomial approximation methods; resulting in estimates that are within 0.01 of the actual value in 90% of the test cases. An arguably even more important measure of the performance of the three techniques is the percentage of cases where the correct worst-case joint failure F is determined. For the 10,000 test cases, the power method was correct 97.5% of the time while the first and second order approximations were correct 33.6% and 35% of the time respectively. [5]
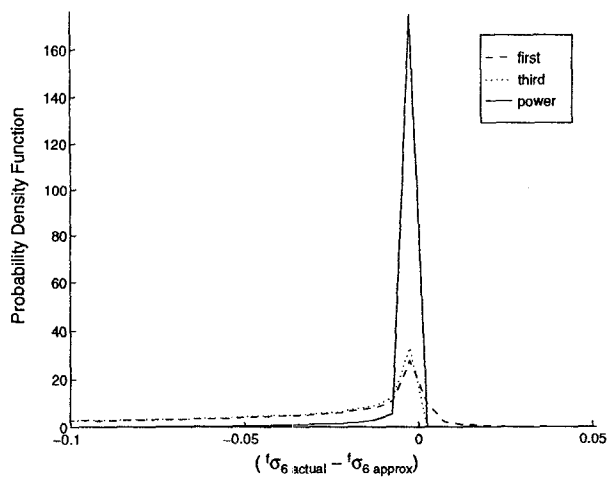

Fig. 2. A comparison of the error in the estimated minimum singular value for the three approximation techniques.

In addition to the above statistical analysis, the performance of the three approximation techniques was evaluated in a real-time control implementation. The familiar three-link planar manipulator with equal link lengths is presented as an illustrative example. The manipulator's end effector is commanded to follow a straight-line trajectory starting from its base and extending to its reach singularity. Fig. 3 shows the joint-space trajectories obtained by using the three techniques along with a contour map of $\mathcal{K}$. The trajectory obtained by using the power method clearly tracks the optimal value of $\mathcal{K}$ and is in fact indistinguishable from the closed-form optimal solution given in [8]. The trajectories obtained by using the polynomial approximation techniques deviate noticeably from the optimal; however, the actual value of $\mathcal{K}$ along these trajectories

---

[4]The magnitude of this perturbation is meant to reflect reasonable values for the maximum joint velocity and the controller cycle time, e.g., one radian per second and ten milliseconds, respectively.

[5]These percentages can be somewhat misleading since in many cases, the minimum singular value after the failure of a joint other than F can be close to $\mathcal{K}$, making the calculation of F ill-conditioned. In such cases, it can be argued that the effect of gradient calculations based on the incorrect F would not be too deleterious.

is quite high, as shown in Fig. 4. Since all three techniques are suitable for real-time control (an unoptimized implementation of the power method required only 0.2 milliseconds on a SPARC 5 to compute $\mathcal{K}$ for a 6×7 Jacobian) the superior accuracy of the power method makes it the obvious method of choice.
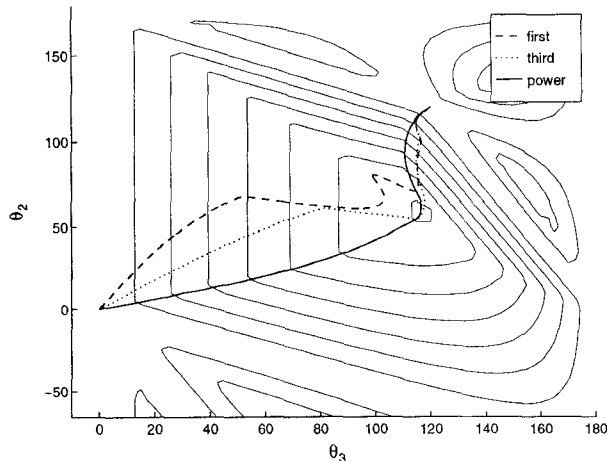

Fig. 3. A comparison of the joint space trajectories obtained using the three approximation techniques for maximizing $\mathcal{K}$. The trajectories are displayed over a contour map of $\mathcal{K}$.
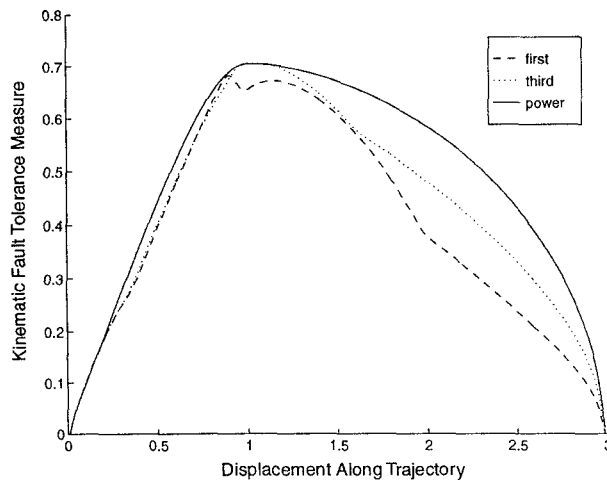

Fig. 4. A comparison of the actual value of $\mathcal{K}$ along the trajectory displayed in Fig. 3.

V. CONCLUSIONS

The main conclusion of this work is that failure tolerance to locked-joint failures is possible in a real-time control scheme. It has been shown that a manipulator can be operated such that at all times it is in an optimal configuration where, should a joint failure occur, the deviation from the desired end-effector trajectory will be minimized and the remaining joints will be able to compensate for the failed one. While three different approaches have been presented for this purpose, the overwhelming method of choice is that based on the power method.

REFERENCES

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

[2] Reliability Analysis Center, *Nonelectronic Parts Reliability Data 1995*, Department of Defense, 201 Mill St., Rome, NY 13440 (315) 337 0900, 1995.

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

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

[5] E Wu, J Hwang, and J Chladek, "Fault tolerant joint development for the space shuttle remote manipulator system:Analysis and experiment", *in Proc. Fourth Int. Symp. on Robot. and Manuf.* , pp. 505–510, Nov. 1992.

[6] D Sreevijayan, S Tosunoglu, and D Tesar, "Architectures for fault-tolerant mechanical systems", *in Proc. of the 7th Mediterranean Electrotechnical Conference*, pp. 1029–1033, Antalya, TURKEY, April 12-14 1994.

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

[8] C L Lewis and A A Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of failures", *Comp. and Elec. Eng.*, pp. 273–288, May 1994.

[9] C J J Paredis, W K F Au, and P K Khosla, "Kinematic design of fault tolerant manipulators", *Comp. and Elec. Eng.*, vol. 20, no. 3, pp. 211–220, May 1994.

[10] D E Whitney, "Resolved motion rate control of manipulators and human prostheses", *IEEE Trans. on Man–Machine Sys.*, vol. MMS-10, no. 2, pp. 47–53, June 1969.

[11] A Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms", *IEEE Trans. on Sys., Man, and Cyb.*, pp. 868–871, December 1977.

[12] J Baillieul, "Kinematic programming alternatives for redundant manipulators", *in Proc. 1985 Int. Conf. on Robot. and Auto.*, pp. 722–728, St. Louis, MO, March 25-28 1985.

[13] O Egeland, "Task-space tracking with redundant manipulators", *IEEE J. Robot. and Auto.*, pp. 471–475, Oct. 1987.

[14] H Seraji, "Configuration control of redundant manipulators: Theory and implementation", *IEEE Trans. on Robot. and Auto.*, vol. 5, no. 4, pp. 472–490, August 1989.

[15] T Yoshikawa, "Manipulability of robotic mechanisms", *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, Summer 1985.

[16] J Angeles, "The design of isotropic manipulator architectures in the presence of redundancies", *Int. J. Robot. Res.*, vol. 11, no.3, pp. 196–201, June 1992.

[17] S L Chiu, "Task compatibility of manipulator postures", *Int. J. Robot. Res.*, vol. 7, no. 5, pp. 13–21, October 1988.

[18] C A Klein and B E Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators", *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 72–83, Summer 1987.

[19] C L Lewis, *Fault Tolerance for Kinematically Redundant Manipulators*, PhD thesis, Purdue University, 1994.

[20] C A Klein and L Chu, "Comparison of extended Jacobian and Lagrange multiplier methods for resolving kinematic redundancy", to appear in *J. Intel. and Robot. Sys.*, 1997.

[21] A A Maciejewski and C A Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations", *J. Robot. Sys.*, vol. 5, no. 6, Dec. 88.

[22] M A Gonzalez-Palacios, J Angeles, and R Ranjbaran, "The kinematic synthesis of serial manipulators with a prescribed Jacobian", *in Proc. 1993 Int. Conf. Robot. and Auto.*, pp 450–455, Atlanta, Georgia, May 2-6 1993.