Short Paper

Real-Time Failure-Tolerant Control of Kinematically Redundant Manipulators

Kenneth N. Groom, Anthony A. Maciejewski, and Venkataramanan Balakrishnan

***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. Comparisons show that the performance of the best method is indistinguishable from that of brute-force implementations. An example demonstrating the real-time performance of the algorithm on a commercially available seven degree-of-freedom manipulator is presented.**

***Index Terms*—Fault/failure tolerance, kinematics, kinematically redundant, locked joint failure, manipulators, redundant robots/manipulators, robots.**

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 h; indeed, 28.7% had mean-time-between-failure of 100 h 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 h of operation (for six-DOF 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 toward 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 alternate 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]–[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.

TABLE I
Example Reliability Data for Typical Robot Components

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

This work presents real-time implementations 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 post-failure joint velocity over all single locked-joint failures. This is also equivalent to maximizing a local measure of the distance to the worst-case post-failure workspace boundary. (The issue of computing a guaranteed post-failure workspace is discussed in [10].) The main contribution of this work is a computationally efficient scheme for computing the fault-tolerance measure and its gradient, which in turn enables real-time optimal configuration of robots in anticipation of failures. In addition, the computational techniques presented here can be used to efficiently identify regions in the workspace with high failure-tolerance.

II. Framework for Failure-Tolerant Control

The forward kinematics of manipulators are frequently represented as

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}} \tag{1}$$

where $\dot{\mathbf{x}}$ is an m-dimensional vector representing the end-effector velocity, $\dot{\mathbf{q}}$ is an n-dimensional vector describing the joint velocities, and J is the m by n manipulator Jacobian matrix [11]. For a redundant manipulator, $n > m$ so the equation is underconstrained and there are an infinite number of solutions which can be expressed as

$$\dot{\mathbf{q}} = J^{+}\dot{\mathbf{x}} + (I - J^{+}J)\mathbf{z} \tag{2}$$

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 [12]. Other methods for optimizing h, such as those based on the extended Jacobian [13] (for tracking critical points) or the augmented Jacobian [14], [15] (for tracking a desired value of h), also require its gradient.

Manuscript received October 26, 1998; revised May 19, 1999. This paper was recommended for publication by Associate Editor J. Tsai and Editor A. De Luca upon evaluation of the reviewers' comments. This work was supported by Sandia National Laboratories under Contract AL-3011. This work was presented at the IEEE International Conference on Robotics and Automation, Albuquerque, NM, April 20–25, 1997.

The authors are with Purdue University, West Lafayette, IN 47907-1285 USA.

Publisher Item Identifier S 1042-296X(99)09111-9.

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

$$J = U\Sigma V^T \tag{3}$$

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 $\Sigma_d = \mathrm{diag}(\sigma_1, \cdots, \sigma_m)$ 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

$$J^+ = V\Sigma^+ U^T \tag{4}$$

as well as the null space projection

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

The SVD has long been a valuable tool for quantifying various dexterity measures: manipulability [16] (product of the singular values), isotropy [17] (ratio of the maximum singular value to the minimum), task compatibility [18] (weighted combination of singular values) and proximity to singularities [19] (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:

$$\mathcal{K} = \min_{f=1\cdots n} \sigma_m({}^fJ) \tag{6}$$

where

$${}^fJ = [\mathbf{j}_1 \quad \mathbf{j}_2 \quad \cdots \quad \mathbf{j}_{f-1} \quad \mathbf{0} \quad \mathbf{j}_{f+1} \quad \cdots \quad \mathbf{j}_n] \tag{7}$$

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

$$F = \arg\min_{f=1\cdots n} \sigma_m({}^fJ). \tag{8}$$

The gradient[1] 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 $\dot{\mathbf{x}}$ [21]. The kinematic fault-tolerance measure can be re-expressed in terms of the singular vectors of the failed Jacobian ${}^FJ$ as

$$\mathcal{K} = {}^F\hat{\mathbf{u}}_m^T\, {}^FJ\, {}^F\hat{\mathbf{v}}_m. \tag{9}$$

The change of $\mathcal{K}$ with respect to q_i is then given by[2]

$$\frac{\partial \mathcal{K}}{\partial q_i} = {}^F\hat{\mathbf{u}}_m^T \frac{\partial {}^FJ}{\partial q_i} {}^F\hat{\mathbf{v}}_m. \tag{10}$$

[1]Note that $\mathcal{K}$ is not differentiable when F is not unique. However, this is addressed through standard numerical techniques [20].

[2]Here we have used the fact that the first order change in a singular vector is orthogonal to the vector itself.

For rotary-jointed robots,[3] the partial derivative of the kth column of the Jacobian with respect to q_i can be efficiently computed as follows [22]:

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

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)

$$\nabla\mathcal{K} = \left[\frac{\partial\mathcal{K}}{\partial q_1}\frac{\partial\mathcal{K}}{\partial q_2}\cdots\frac{\partial\mathcal{K}}{\partial q_n}\right]^T. \tag{12}$$

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

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.

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. (Note that the full SVD can be computed efficiently using Givens rotations [23].)

A. First Order Polynomial Approximation

Let

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

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

[3]Prismatic joints simplify the expressions.

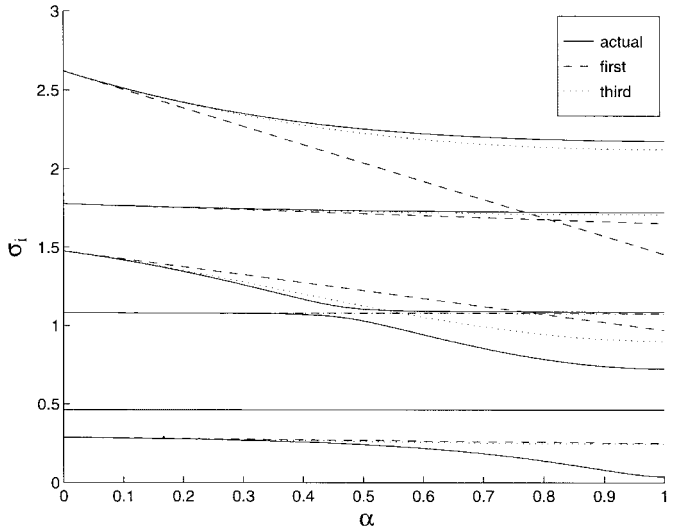

Fig. 1. An example of the change in singular values of a typical $J_{6\times 7}$ as a column is removed.

approximation of the the singular values for ${}^f J(1)$ can be obtained from the SVD of $J(0)$ as

$$ {}^f\sigma_i(1) = {}^f\sigma_i(0) + \left.\frac{d\,{}^f\sigma_i}{d\alpha}\right|_{\alpha=0} \tag{14} $$

where

$$ \frac{d\,{}^f\sigma_i}{d\alpha} = {}^f\hat{\mathbf{u}}_i^T \frac{d\,{}^f J}{d\alpha}\,{}^f\hat{\mathbf{v}}_i \tag{15} $$

and

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

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

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

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.[4] We use (17) 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 ${}^F J$, required to calculate the gradient from (10).

The accuracy of the first-order approximation of 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 "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.

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

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

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

$$ \frac{d\hat{\mathbf{v}}_i}{d\alpha} = (1-\alpha)\hat{\mathbf{u}}_i^T\mathbf{j}_f \sum_{\substack{k=1\\k\neq i}}^{m} \frac{\hat{\mathbf{u}}_k^T\mathbf{j}_f\left(\frac{\sigma_i}{\sigma_k}+\frac{\sigma_k}{\sigma_i}\right)}{\sigma_k^2-\sigma_i^2}\hat{\mathbf{v}}_k - \hat{\mathbf{u}}_i^T\mathbf{j}_f \sum_{k=m+1}^{n} \frac{v_{fk}}{\sigma_i}\hat{\mathbf{v}}_k. \tag{19} $$

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 $d^2\,{}^f J/d\alpha^2 = 0$), we have

$$ \frac{d^2\,{}^f\sigma_i}{d\alpha^2} = \frac{d\,{}^f\hat{\mathbf{u}}_i^T}{d\alpha}\frac{d\,{}^f J}{d\alpha}\,{}^f\hat{\mathbf{v}}_i + {}^f\hat{\mathbf{u}}_i^T\frac{d\,{}^f J}{d\alpha}\frac{d\,{}^f\hat{\mathbf{v}}_i}{d\alpha}. \tag{20} $$

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

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

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 ${}^f\sigma_i(\alpha)$:

$$ {}^f\sigma_i(\alpha) = \sigma_i(0) + {}^f\sigma_i'(0)\alpha + \tfrac{1}{2}\,{}^f\sigma_i''(0)\alpha^2 - \tfrac{1}{3}({}^f\sigma_i'(0) + {}^f\sigma_i''(0))\alpha^3 \tag{22} $$

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 [24]. 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 ${}^{F}J$. 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 $({}^{f}J\,{}^{f}J^{T})^{-1}$. We write

$$ {}^{f}J\,{}^{f}J^{T} = JJ^{T} - \mathbf{j}_f\mathbf{j}_f^{T} \tag{23} $$

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

$$ ({}^{f}J\,{}^{f}J^{T}) = U(\Sigma\Sigma^{T} - U^{T}\mathbf{j}_f\mathbf{j}_f^{T}U)U^{T}. \tag{24} $$

Therefore

$$ U^{T}({}^{f}J\,{}^{f}J^{T})^{-1}U = (\Sigma\Sigma^{T} - U^{T}\mathbf{j}_f\mathbf{j}_f^{T}U)^{-1}. \tag{25} $$

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

$$ {}^{f}\hat{\mathbf{u}}_i = U\,{}^{f}\hat{\mathbf{w}}_i \tag{26} $$

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

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

we get

$$ \begin{aligned} &(\Sigma\Sigma^{T} - U^{T}\mathbf{j}_f\mathbf{j}_f^{T}U)^{-1} \\ &\quad = (\Sigma\Sigma^{T})^{-1} + \frac{((\Sigma\Sigma^{T})^{-1}U^{T}\mathbf{j}_f)(\mathbf{j}_f^{T}U(\Sigma\Sigma^{T})^{-1})}{1 - \mathbf{j}_f^{T}U(\Sigma\Sigma^{T})^{-1}U^{T}\mathbf{j}_f}. \end{aligned} \tag{28} $$

We have assumed that $\Sigma\Sigma^{T}$ is nonsingular;[5] otherwise, we know that $\mathcal{K} = 0$. Noting that

$$ \mathbf{j}_f^{T}U = [\sigma_1 v_{f1} \quad \sigma_2 v_{f2} \quad \cdots \sigma_m v_{fm}] \tag{29} $$

we get

$$ \begin{aligned} (\Sigma\Sigma^{T} - U^{T}\mathbf{j}_f\mathbf{j}_f^{T}U)^{-1} &= \operatorname{diag}(1/\sigma_1^2, \cdots, 1/\sigma_m^2) \\ &\quad + \frac{1}{\sum_{i=m+1}^{n} v_{fi}^2} \begin{bmatrix} v_{f1}/\sigma_1 \\ \vdots \\ v_{fm}/\sigma_m \end{bmatrix} \cdot [v_{f1}/\sigma_1 \cdots v_{fm}/\sigma_m]. \end{aligned} \tag{30} $$

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

$$ \begin{aligned} {}^{f}\mathbf{w}_m &:= (U^{T}({}^{f}J\,{}^{f}J^{T})^{-1}U)\,{}^{f}\hat{\mathbf{w}}_m; \\ {}^{f}\sigma_m &:= ||{}^{f}\mathbf{w}_m||; \\ {}^{f}\hat{\mathbf{w}}_m &:= {}^{f}\mathbf{w}_m/{}^{f}\sigma_m. \end{aligned} $$

[5]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 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 $({}^{f}J\,{}^{f}J^{T})^{-1}$ 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 associated left singular vector ${}^{F}\hat{\mathbf{u}}_m$ can be computed using (26); the right singular vector can be computed using

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

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 [25]. The previous configuration for this manipulator is then determined by applying a perturbation of 0.01 rads to each of the joints.[6] 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.

Fig. 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 0.0005 of the actual value in 98% 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 99.98% of the time while the first and third order approximations were correct 33.6% and 35% of the time, respectively.[7]

[6]The magnitude of this perturbation is meant to reflect reasonable values for the maximum joint velocity and the controller cycle time, e.g., 1 rad/s and 10 msec, respectively.

[7]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. However, this is not a problem as our algorithm uses a combination of the gradients corresponding to the two worst-case joint failures.

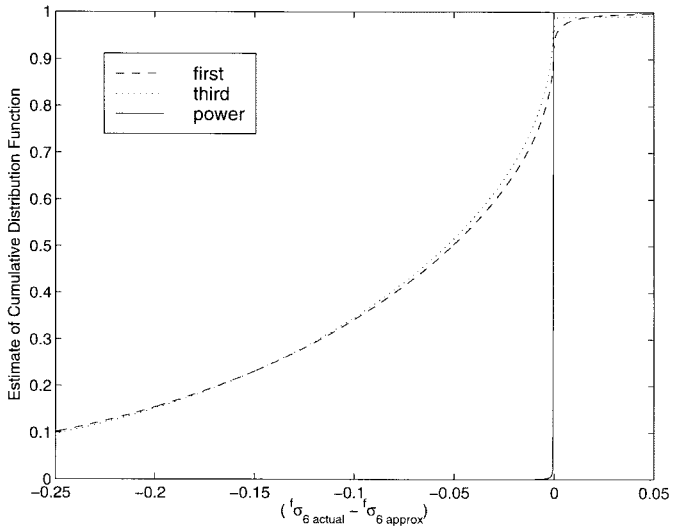

Fig. 2. Comparison of the error in the estimated minimum singular value for the three approximation techniques.

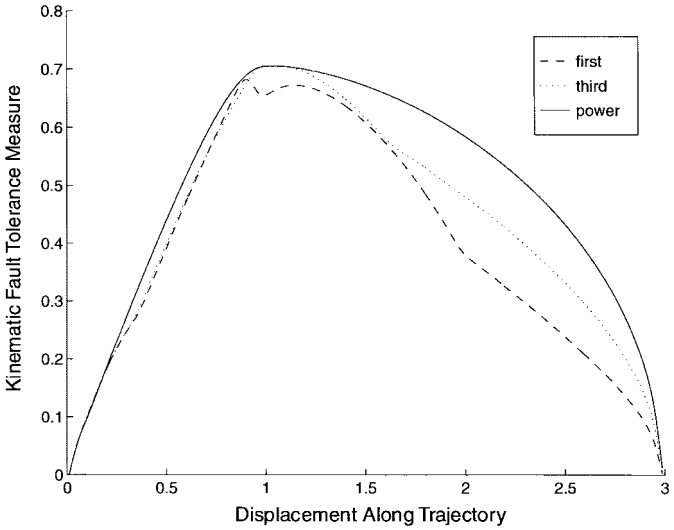

Fig. 4. Comparison of the actual value of $\mathcal{K}$ along the trajectory displayed in Fig. 3.

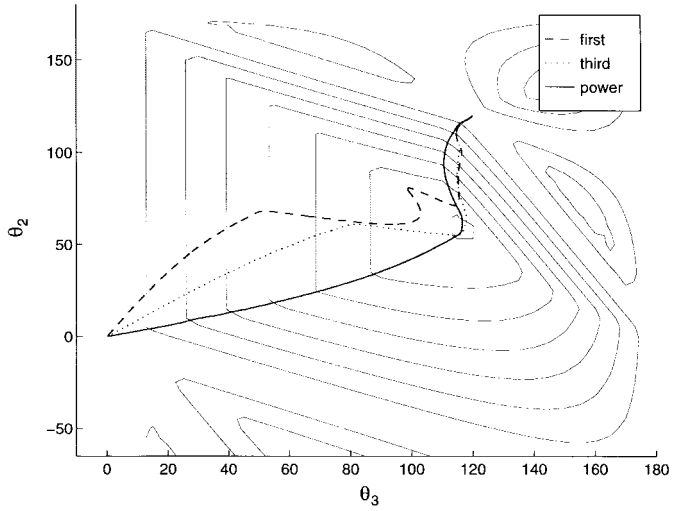

Fig. 3. Comparison of the joint space trajectories obtained using the three approximation techniques for maximizing $\mathcal{K}$. The trajectories are displayed over a contour map of $\mathcal{K}$.

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 is quite high, as shown in Fig. 4. Since all three techniques are suitable for real-time control the superior accuracy of the power method makes it the obvious method of choice.

V. A Specific Seven-DOF Example

The proposed fault-tolerance control scheme is now demonstrated on a task requiring three dimensional positioning and orienting of the end-effector. The manipulator chosen for the example is the Robotics Research Corporation K-1207i, a commercially available spatially redundant manipulator, whose Denavit-Hartenberg parameters (Paul's convention [26]) are shown in Table II. The parameters of length are given in meters in the table; the linear portion of the Jacobian was scaled by dividing by 0.3 m, which is consistent with the characteristic length [27] of the manipulator, rendering it very close to isotropic (condition number 1.43) for a tool offset of $\theta_8 = 4.50$ rad and $d_9 = 0.34$ m.

TABLE II
Denavit–Hartenberg Parameters and Joint Limits for the RRC K-1207i Arm. The Last Two Rows Correspond to the Parameters for a Tool Offset. Lengths in Meters, Angles in Radians

Joint	a	d	α	θ	θ min	θ max
1	-0.102	0.000	-1.570	θ_1	-3.141	3.141
2	0.102	0.000	1.570	θ_2	-3.051	-0.054
3	-0.086	0.546	-1.570	θ_3	-6.280	0.000
4	0.086	0.000	1.570	θ_4	-3.051	0.000
5	-0.059	0.546	-1.570	θ_5	-6.280	6.280
6	0.059	0.000	1.570	θ_6	-2.967	0.610
7	0.000	0.178	-1.570	θ_7	-6.280	6.280
8	0.000	0.000	1.570	θ_8		
9	0.000	d_9	0.000	0.000		

In this example, we consider several issues associated with failure tolerance. The first is the design of the offset between the end-effector coordinate frame and the wrist mounting plate. The second is mapping the task into an optimally failure-tolerant workcell location. (We assume that errors in all six task dimensions are equally important.) The third is the actual operation of the manipulator for the prescribed task in an optimally failure-tolerant manner, and comparing its performance to pseudoinverse control (the latter is used as a representative example). The first two issues are design problems and can, therefore, be performed off-line, however, the third requires that the calculations be performed in real time.

The first two issues can be considered simultaneously by maximizing $\mathcal{K}$ over the manipulator's configuration (the vector $\theta = [\theta_1 \cdots \theta_7]$) and the tool-offset parameters (θ_8 and d_9). For convenience, we define $\mathbf{q} = [\theta_1 \cdots \theta_8 \quad _9]$. Unfortunately, maximization of $\mathcal{K}$ is a very difficult problem, owing to the fact that $\mathcal{K}$ has a large number

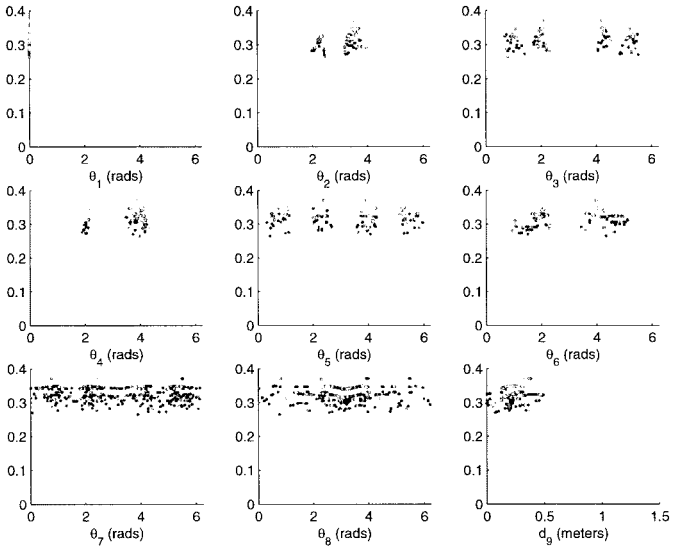

Fig. 5. $\mathcal{K}$ of the best 1000 configurations out of 20 mil randomly generated manipulator configurations after gradient optimization. Each configuration is shown using a different shade of gray. Note that the value of $\mathcal{K}$ is independent of θ_1.

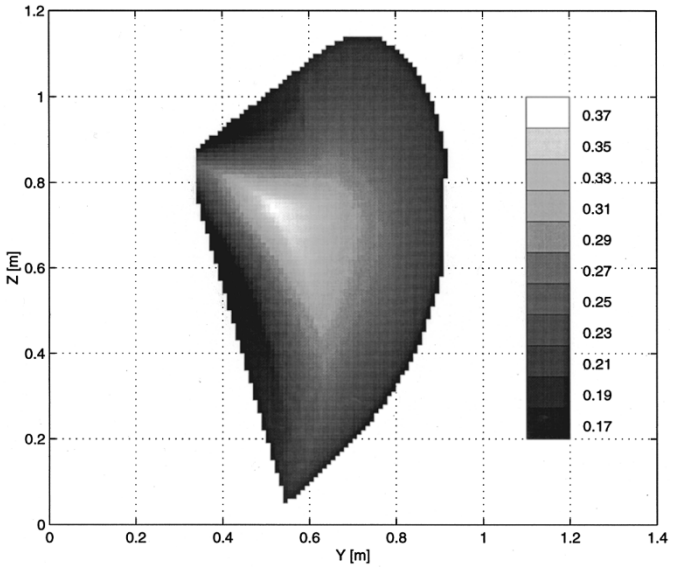

Fig. 6. Failure tolerance of the workspace surrounding the optimal point. The robot's base coordinate frame is located at [0.0 0.0 0.81]m and the axis of rotation for joint one is oriented along Z. Since $\mathcal{K}$ is independent of θ_1, these values are rotationally symmetric about Z. Orientation of the end-effector was held constant during mapping.

of local maxima. The dimension of the search space precludes an exhaustive search method, such as gridding. We therefore applied a standard Monte Carlo method to perform the maximization.

We first evaluated $\mathcal{K}$ at 20 mil randomly generated values of $\mathbf{q}$. The average value of $\mathcal{K}$ is 0.02, indicating that disregarding possible failures will likely result in severely degraded post-failure performance. To maximize our fault-tolerance measure, the values of $\mathbf{q}$ corresponding to the top 1000 values of $\mathcal{K}$ were chosen as initial points for a simple gradient ascent algorithm. This resulted in 185 distinct local maxima, with $\mathcal{K}$ values ranging between 0.27 and 0.37. These are illustrated in Fig. 5. Note that despite the fact that a randomly selected configuration will likely have low $\mathcal{K}$, there do exist a large number of configurations and tool-offset parameter values that have relatively high $\mathcal{K}$. Moreover, fixing the manipulator configuration and tool-offset parameters to those corresponding to the global maximum

$$\mathbf{q} = [2.43 \quad 3.47 \quad 4.24 \quad 3.95 \quad 2.43 \quad 4.01 \quad 5.51 \quad 3.96 \quad 0.35]^T$$

and varying the configuration θ of the manipulator reveals that relatively high $\mathcal{K}$ can be achieved over a fairly large portion of the workspace (see Fig. 6). In general, an analysis such as the above can be employed to first identify regions in the workspace with good fault-tolerance. Other factors that affect workpiece location such as collision avoidance and joint limits can then be incorporated to arrive at the final workpiece location.

To illustrate the performance of our real-time fault-tolerant control, we selected as an example task the painting of a 0.3 m $\times$ 0.4 m plate with the K-1207i manipulator. Taking all design considerations into account, the configuration

$$\mathbf{q} = [2.41 \quad 3.63 \quad 4.31 \quad 4.10 \quad 2.54 \quad 4.23 \quad 5.05 \quad 1.57 \quad 0.1]^T$$

with $\mathcal{K} = 0.31$ was selected to map the paint gun to the center of the plate. (The task and the selected manipulator configuration are shown in Fig. 7.) The K-1207i is commanded to follow the indicated trajectory with an end-effector speed of 0.085 m/sec. At precisely the center of the plate, when the paint gun is making an upward vertical motion (along the positive $\hat{\mathbf{z}}$ direction), a joint of the manipulator is failed and locked into place. The controller responds to the failure

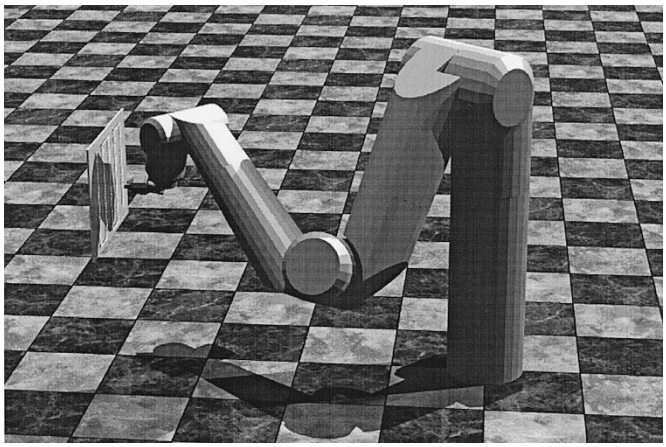

Fig. 7. Example task for demonstrating failure tolerance. The manipulator is painting the flat plate using vertical motions of the paint gun. At precisely the center of the plate, a locked joint failure occurs.

by removing the appropriate column of the Jacobian and continuing the task with the remaining joints.

As a baseline for comparison, the task was performed using standard pseudoinverse control,[8] with the arm configuration at the point of failure being

$$\theta = [1.78 \quad 5.32 \quad 5.98 \quad 4.48 \quad 0.40 \quad 6.05 \quad 5.43]^T.$$

Fig. 8 illustrates the effect of a joint six failure on joint velocity while using pseudoinverse control. At the point of failure, large discontinuities are experienced as the functioning joints accelerate to compensate for the lost joint. (Note that no dynamics have been incorporated into the simulation for simplicity of illustration. In reality, the manipulator dynamics will result in the acceleration being finite, which will manifest itself in additional end-effector tracking error.) The size of the discontinuities can be inferred from the SVD of

[8] Note that *any* inverse kinematic solution which does not configure the arm in anticipation of a failure can produce similar undesirable behavior.

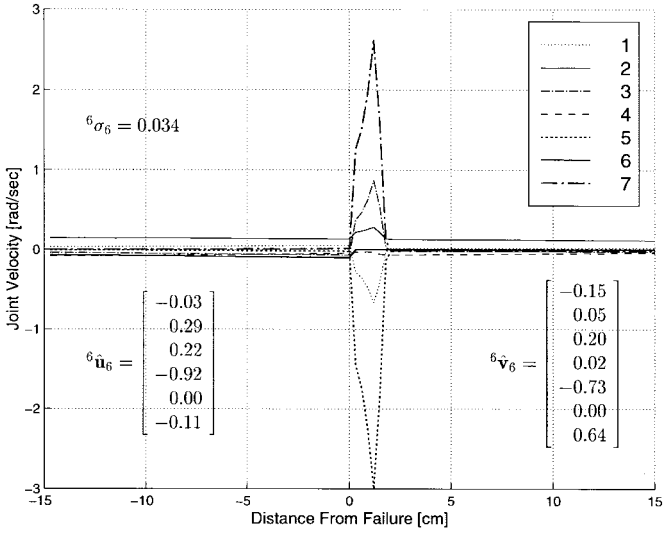

Fig. 8. Joint velocity during joint six failure while using pseudoinverse control.

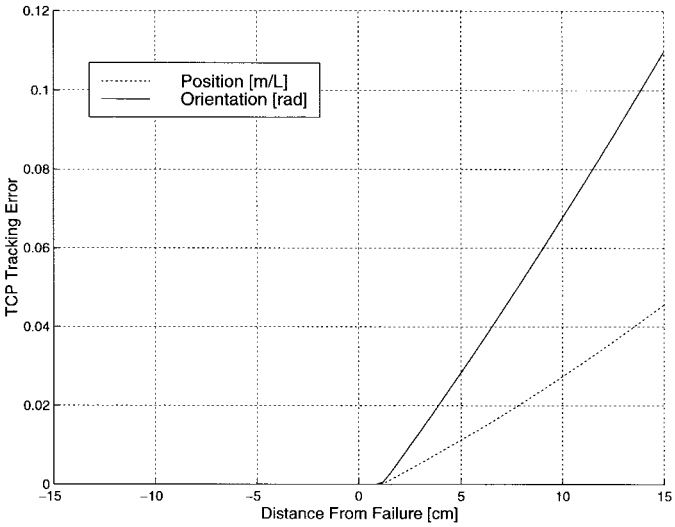

Fig. 9. End-effector error during joint six failure while using pseudoinverse control (The scaling factor on the translational error is given by L = 0.3 m).

the post-failure Jacobian: The minimum singular value $^6\sigma_6$ is 0.034, with the corresponding singular vectors of

$$^6\hat{\mathbf{u}}_6 = [-0.03 \quad 0.29 \quad 0.22 \quad -0.92 \quad 0.00 \quad -0.11]^T$$

and

$$^6\hat{\mathbf{v}}_6 = [-0.15 \quad 0.05 \quad 0.20 \quad 0.02 \quad -0.73 \quad 0.00 \quad 0.64]^T.$$

The small value of $^6\sigma_6$ indicates that the manipulator has low dexterity along $^6\hat{\mathbf{u}}_6$. The projection of the desired end-effector velocity on $^6\hat{\mathbf{u}}_6$ is simply its third element, i.e., 0.22, scaled by $||\dot{\mathbf{x}}||$. Thus the desired end-effector velocity is far from being the "worst-case;" however, the very low value of $^6\sigma_6$ results in large joint velocities along $^6\hat{\mathbf{v}}_6$, consistent with the discontinuities observed in Fig. 8. After the initial discontinuity, there is a further rise in the joint velocities, indicating that the manipulator is approaching a singularity. The subsequent sharp drop in the joint velocities then corresponds to the manipulator being at a post-failure workspace boundary, and no longer being able to track the desired trajectory (see Fig. 9).

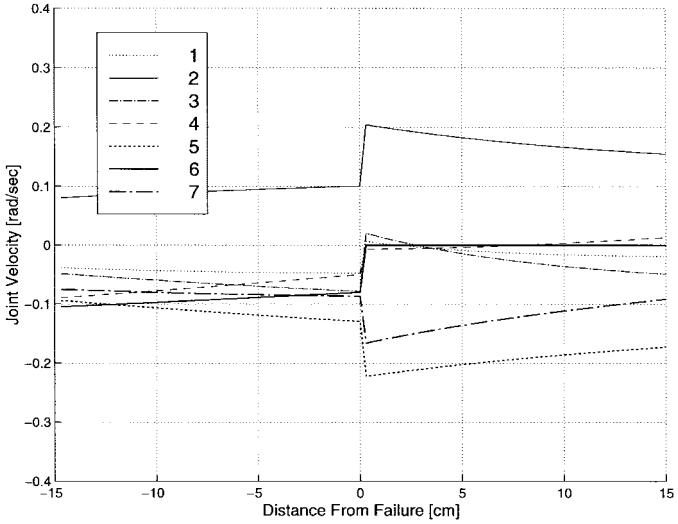

Fig. 10. Joint velocity during joint six failure while using failure-tolerant control.

Fig. 10 shows the joint velocities with failure-tolerant control.[9] Note the markedly lower discontinuity in the joint velocity at failure, indicating that the remaining joints are able to compensate for the failed joint with more practical acceleration requirements. In addition, as suggested by the slowly varying joint velocities, the manipulator is able to complete the task.

Indeed, since the minimum singular value of a matrix is the best measure of its proximity to singularity, maximizing $\mathcal{K}$ is equivalent to locally maximizing the distance to the post-failure workspace boundaries.

We should point out that the example presented here is a representative one; similar behavior is encountered with the failure of other joints as well. Furthermore, it can be shown that for every end-effector position, the manipulator can be in a configuration with zero $\mathcal{K}$. Details can be found in [20].

VI. Conclusion

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.

References

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

[2] R. A. Center, *Nonelectronic Parts Reliability Data 1995*. Rome, NY: Department of Defense, 1995.

[3] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 477–490, Aug. 1995.

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

[5] E. Wu, J. Hwang, and J. Chladek, "Fault tolerant joint development for the space shuttle remote manipulator system: Analysis and experiment," in *Proc. 4th Int. Symp. Robot. Manufact. (ISRAM '92)*, Sante Fe, NM, Nov. 11–13 1992, pp. 505–510.

[9]The average computation time for the fault-tolerant inverse kinematic solution was 1.1 ms on a SUN Sparcstation 20 with a 60 MHz processor. The algorithm was implemented in C using double-precision arithmetic, without any special code optimization.

[6] D. Sreevijayan, S. Tosunoglu, and D. Tesar, "Architectures for fault-tolerant mechanical systems," in *Proc. 7th Mediterranean Electrotech. Conf.*, Antalya, Turkey, Apr. 12–14 1994, pp. 1029–1033.
[7] A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," in *Proc. 1990 Int. Conf. Robot. Automat.*, Cincinnati, OH, May 13–18 1990, pp. 638–642.
[8] C. L. Lewis and A. A. Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of failures," *Comput. Electron. Eng.*, vol. 20, no. 3, pp. 273–288, May 1994.
[9] C. J. J. Paredis, W. K. F. Au, and P. K. Khosla, "Kinematic design of fault tolerant manipulators," *Comput. Electron. Eng.*, vol. 20, no. 3, pp. 211–220, May 1994.
[10] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 622–629, Aug. 1997.
[11] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man–Mach. Syst.*, vol. MMS-10, pp. 47–53, June 1969.
[12] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 868–871, Dec. 1977.
[13] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. 1985 Int. Conf. Robot. Automat.*, St. Louis, MO, Mar. 25–28 1985, pp. 722–728.
[14] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 471–475, Oct. 1987.
[15] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 472–490, Aug. 1989.
[16] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.
[17] 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.
[18] S. L. Chiu, "Task compatibility of manipulator postures," *Int. J. Robot. Res.*, vol. 7, no. 5, pp. 13–21, Oct. 1988.
[19] 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.
[20] K. N. Groom, *Real-Time Failure-Tolerant Control of Kinematically Redundant Robotic Manipulators*, Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1997.
[21] C. L. Lewis, *Fault Tolerance for Kinematically Redundant Manipulators*, Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1994.
[22] C. A. Klein and L. Chu, "Comparison of extended Jacobian and Lagrange multiplier based methods for resolving kinematic redundancy," *J. Intell. Robot. Syst.*, vol. 19, no. 1, pp. 39–54, May 1997.
[23] A. A. Maciejewski and J. M. Reagin, "A parallel algorithm and architecture for the control of kinematically redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 405–414, Aug. 1994.
[24] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Robot. Syst.*, vol. 5, no. 6, pp. 527–552, Dec. 1988.
[25] 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. Automat.*, Atlanta, GA, May 2–6 1993, pp. 450–455.
[26] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, MA: MIT Press, 1981.
[27] M. Tandirci, J. Angeles, and F. Ranjibaran, "The characteristic point and the characteristic length of robotic manipulators," in *Proc. Robot., Spatial Mech., Mech. Syst.*, Scottsdale, AZ, Sept. 13–16 1992, pp. 203–207.

Self Calibration of Stewart–Gough Parallel Robots Without Extra Sensors

Wisama Khalil and Sébastien Besnard

***Abstract*—This paper presents a method for the autonomous calibration of six degrees-of-freedom parallel robots. The calibration makes use of the motorized prismatic joint positions corresponding to some sets of configurations where in each set either a passive Universal joint or a passive Spherical joint is fixed using a lock mechanism. Simulations give us an idea about the number of sets that must be used, the number of configurations by set and the effect of noise on the calibration accuracy. The main advantage of this method is that it can be executed rapidly without need to external sensors to measure the position or the orientation of the mobile platform.**

***Index Terms*—Calibration, identification, optimization, parallel robot, static accuracy.**

I. Introduction

The aim of the kinematic calibration is to calculate accurately the values of the kinematic parameters of the robot in order to improve its accuracy.

The classical methods for parallel robot calibration need external sensors to measure the position and orientation of the mobile platform [1]–[4]. The calibration problem is formulated in terms of a measurement residual that is the difference between the measured and computed motorized joint variables, it uses the inverse kinematic model that is easy to calculate for parallel robots.

Self calibration methods using extra sensors on the passive joints have been also proposed for parallel robots.

1) The method presented by Wampler *et al*. [5] for Stewart–Gough robot is based on using position sensors on the 5 passive joints of one leg.
2) In the method of Zhuang and Liu [6], a limited number of passive Universal joints are needed to be measured.
3) Nahvi *et al*. [7] presented an autonomous calibration method for a shoulder-joint robot with three degrees of freedom (d.o.f.) but particular geometry with double joints on each platform point must be verified.
4) Zhuang [8] has presented new methods based on the use of extra sensors on some passive Universal joints.

The calibration methods based on redundant sensors on the passive joints adjust the values of the kinematic parameters in order to minimize a residual between the measured and the calculated values of the angles of these joints. In order to get appropriate accuracy for the identified parameters, big accuracy is needed on these sensors. Putting sensors on an already manufactured robot is not a trivial problem, it must be foreseen while designing the robot.

It is to be noted that redundant sensors on the U-joints have been proposed also to get an analytical solution of the direct kinematic model [9]–[11]. But in this case moderate accuracy is sufficient.

The calibration method presented in this paper uses only the variables of the motorized prismatic joints corresponding to configurations where either one Universal joint or one Spherical joint is

Manuscript received December 15, 1998. This paper was recommended for publication by Associate Editor H. Zhuang and Editor A. De Luca upon evaluation of the reviewers' comments.

The authors are with the Institut de Recherche en Cybernétique de Nantes, Nantes Cedex 44 321, France (e-mail: Wisama.Khalil@ircyn.ec-nantes.fr).

Publisher Item Identifier S 1042-296X(99)09232-0.