Tolerance Volume Due to Joint Variable Errors in Robots

Woo-Jong Lee
Tony C. Woo

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

May 1988
Technical Report 88-5

To appear in the *ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design.*

Abstract

The locational uncertainty of a manipulator is largely due to the errors of the joint variables. But these errors can not be easily compensated for because they are dependent on the operation (i.e., robot-configuration). Motivated by the need to conduct precision engineering and the intellectual curiosity of geometric uncertainty, the probabilistic tolerance volume due to joint errors is investigated.

By defining the locational uncertainty in Cartesian space as a tolerance volume, the investigation focuses on the automatic generation of the tolerance volume from a given confidence level. For this purpose, the linear mapping from $\Delta\mathbf{q}$ space to $\Delta\mathbf{d}$ space through Jacobian matrix is analyzed probabilistically. Probabilistic approach is advantageous since the tolerance volume by the deterministic approach is found to be unnecessarily large. With the assumption of normality of joint variables, this paper begins with the computation of the confidence level for a given tolerance volume. A fast analytic procedure, which gives a considerable time-reduction compared to the commonly used Monte-Carlo simulation, is presented. Based on the monotonic relation between confidence level and tolerance volume, the procedure is used to generate the tolerance volume covering the desired confidence level. The scheme is tested with the six degrees-of-freedom Stanford manipulator and shows a significant (more than 5 times) reduction in the size of the tolerance volume with a 0.3% probability of error.

1. Introduction

In the study of robotics, the location of a manipulator in the three-dimensional Cartesian space is described by two attributes: position and orientation. Since position and orientation are each defined by three degrees-of-freedom (DOF), a location is completely defined by six DOF. The location of a manipulator is effected by kinematic parameters [1,12]. However, the actual location is often somewhat different from the desired location due to errors in the kinematic parameters.

The location of an end-effector is determined by the accuracy of its kinematic parameters, of which there are four: three being the geometric link parameters and one being the joint variable. Error in the geometric link parameters is due to the variability in the machining of the geometric links. This source of kinematic error is fixed as soon as the links are assembled. It can be identified by calibration [5,9,14] and compensated for. Error in the joint variable is of a larger magnitude than those in the geometric link parameters [15]. It is due to a number of control-related factors as well as bearing clearances and is dependent on the particular robot configuration in space. As there are many possible robot configurations, the "randomness" of the joint error is examined in this paper.

Joint error and locational error are defined respectively as the deviations from the desired (nominal) joint value and the corresponding location. They are represented in terms of the differential joint variable vector $\Delta\mathbf{q}$ and the differential location vector $\Delta\mathbf{d}$. The relationship between these two vectors is described by the Jacobian matrix $\mathbf{J}$ [12]:

$$\Delta\mathbf{d} = \mathbf{J}\,\Delta\mathbf{q} \tag{1}$$

For an n DOF manipulator, $\Delta\mathbf{q}$ is a $n\times 1$ vector which consists of the n differential joint variables $\Delta q_1,\ldots,\Delta q_n$.† The differential location vector $\Delta\mathbf{d}$ is always a 6×1 vector, composed of the 3×1 differential position vector and the 3×1 differential orientation vector. The Jacobian matrix $\mathbf{J}$ is then a $6\times n$ matrix and

† The variable for the j-th joint is represented by q_j in this paper. For a revolute joint j, q_j is the joint angle parameter. For a prismatic joint j, q_j is the link length parameter.

it represents the infinitesimal displacement of the end-effector due to an infinitesimal change in a joint variable. Equation (1) describes the linear mapping from an n-dimensional vector space $\mathbf{R}^n$ to a six-dimensional vector space $\mathbf{R}^6$. This mapping is illustrated in Figure 1.

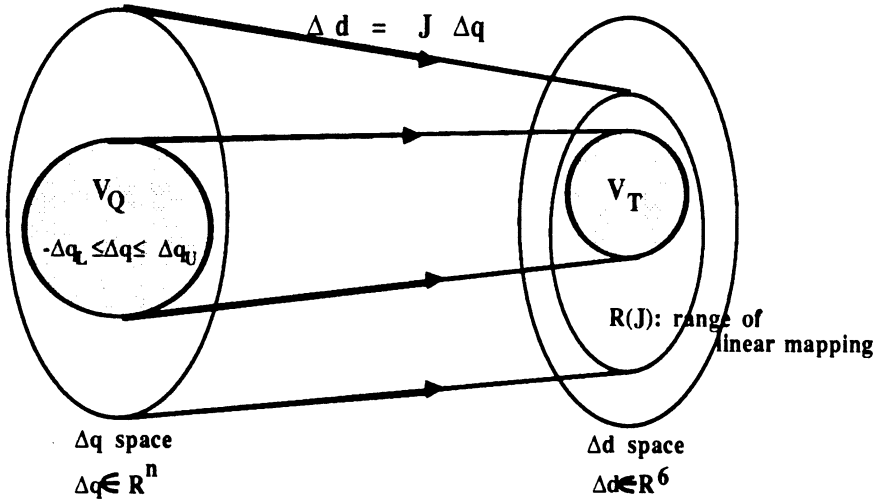


Figure 1. Linear Mapping by Jacobian Matrix

If the joint error were fixed, the locational error could be compensated for by equation (1). But because the joint error is configuration dependent, it is difficult to prescribe all the possible errors. Usually the joint error vector is specified by upper and lower limits as:

$$-\Delta q^L \le \Delta q \le \Delta q^U \tag{2}$$

The locational error due to joint errors (2) is then described by a certain volume in the six-dimensional space, the three dimensions of which are for position errors and the other three are for orientation errors. This volume is referred to as the *tolerance volume* $\mathbf{V}_T$ and the procedure for computing the tolerance volume is referred to as *joint error analysis.*

Joint error analysis examines the tolerance volume $\mathbf{V}_T$ in the Cartesian space, as a linear mapping of the domain $\mathbf{V}_Q$ (in the $\Delta\mathbf{q}$ space). This analysis is relatively simple to perform if the joint errors $\Delta\mathbf{q}$ are

treated as deterministic variables. By so doing, the result of the analysis is for the worst case. To assess the merit of a less pessimistic solution, the probabilistic approach is taken in this paper.

For comparison, the two approaches, deterministic and probabilistic, are examined here by taking a two DOF planar manipulator of Figure 2 as an example. In probabilistic terms, the domain V_Q and its range V_T are understood as confidence intervals. The kinematic equation relating the end-effector position (x,y) to the joint displacement (q_1,q_2) are given by

$$\begin{aligned} x(q_1,q_2) &= l_1 \cos q_1 + l_2 \cos(q_1+q_2) \\ y(q_1,q_2) &= l_1 \sin q_1 + l_2 \sin(q_1+q_2) \end{aligned} \tag{3}$$

From equation (3), the Jacobian matrix of the planar manipulator can be computed as follows:

$$J = \begin{bmatrix} \frac{\partial x(q_1,q_2)}{\partial q_1} & \frac{\partial x(q_1,q_2)}{\partial q_2} \\ \frac{\partial y(q_1,q_2)}{\partial q_1} & \frac{\partial y(q_1,q_2)}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin (q_1+q_2) & -l_2 \sin (q_1+q_2) \\ l_1 \cos q_1 + l_2 \cos (q_1+q_2) & -l_2 \cos (q_1+q_2) \end{bmatrix}$$

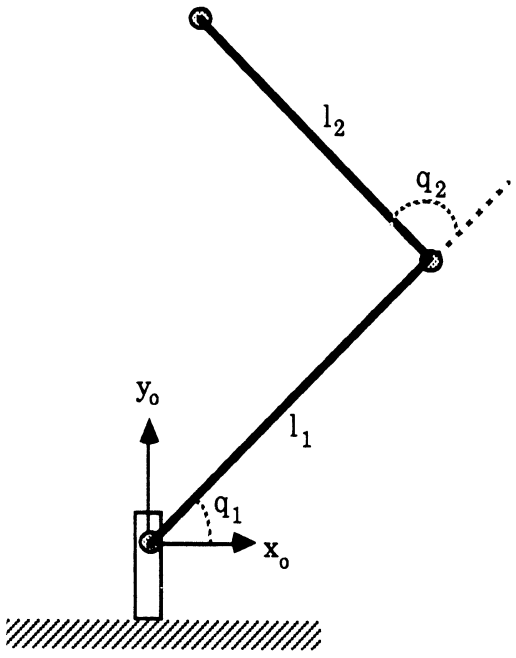


Figure 2. Two DOF Planar Manipulator

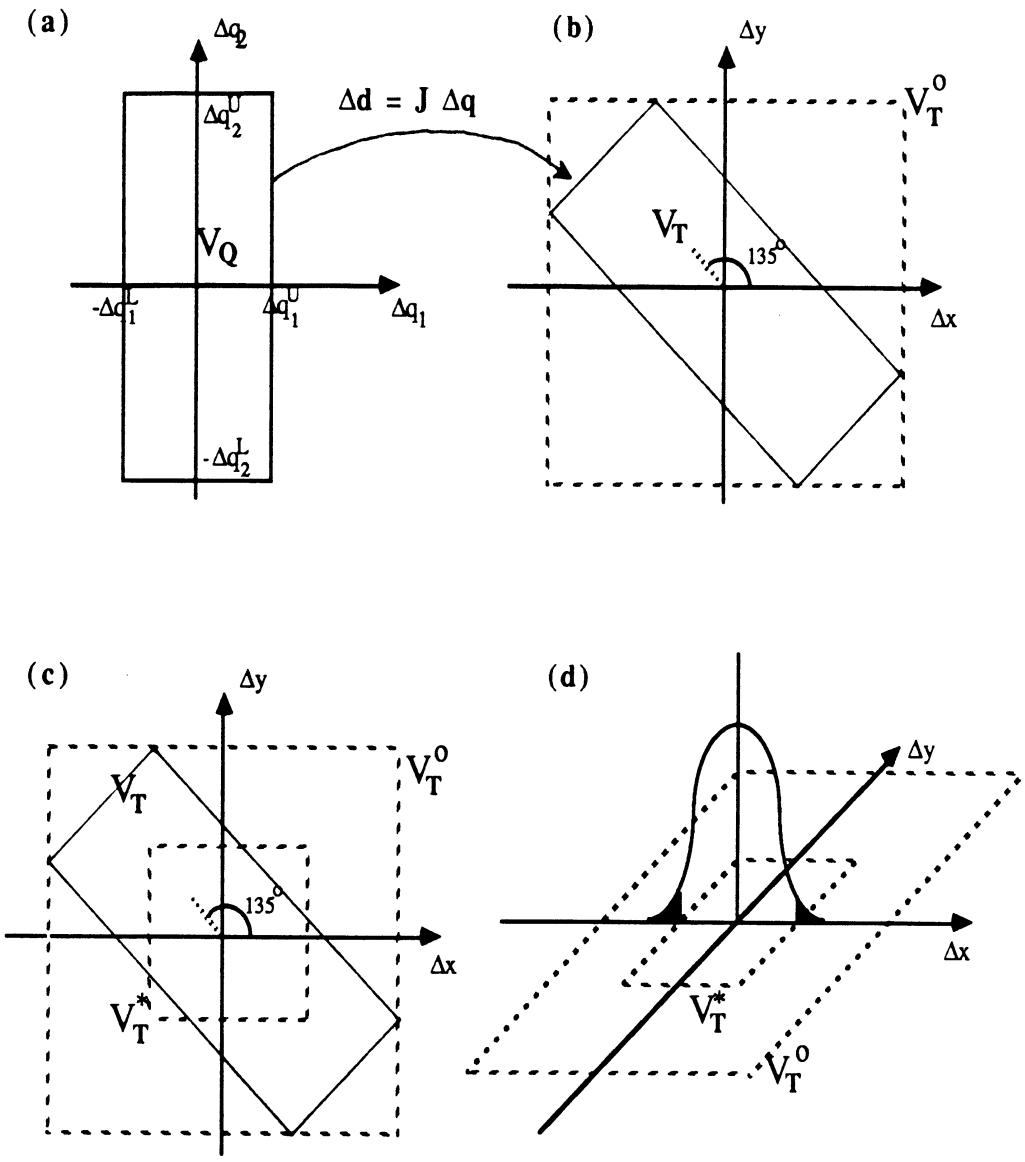


Figure 3. Linear Mapping of a Planar Manipulator

Suppose that the link variables are l_1=1.0 and l_2=1.4142, and the joint variables are $q_1=0^o$ and $q_2=135^o$.

Then,

$$J = \begin{bmatrix} -0.7071 & -0.7071 \\ 0.7071 & -0.7071 \end{bmatrix}$$

The linear mapping by this Jacobian matrix corresponds to a rotation of the set V_Q by an amount of 135^o in the counter-clockwise direction around the origin. See Figure 3-(a) and 3-(b).

For this computation, Benhabib, et al [3] obtained a tolerance volume by taking the absolute values of the Jacobian elements and the maximum absolute joint errors in equation (1). That is, the differential location vector due to the joint errors of equation (2) is specified by the following inequalities.

$$-\Delta d^*_i \le \Delta d_i \le \Delta d^*_i \qquad \text{for } 1 \le i \le 6 \tag{4}$$

where the limit value Δd^*_i is obtained by the following equation

$$\Delta d^*_i = \sum_{j=1}^{n} |J_{ij}| \, \Delta q^*_j \qquad \text{for } 1 \le i \le 6 \tag{5}$$

where $\Delta q^*_j = \max(\Delta q^L_j, \Delta q^U_j)$. Using the same example, [3] produced the tolerance volume V^o_T in Figure 3-(b). The resulting tolerance volume V^o_T from the deterministic approach of [3] covers V_T completely. But, by permitting a small fraction of non-coverage, say, 0.3%, it may be possible to obtain a tolerance volume V^*_T much smaller than V^o_T as illustrated in Figures 3-(c) and 3-(d). (The intuition here is to sacrifice the tail of a probability density function, which has a large range in the abscissa yet the area under it, i.e., the probability, is very small.) To examine the potential pay-off, the probabilistic approach which treats the joint variables as normally distributed random variables is adopted in this paper. The probabilistic approach of this paper differs from other probabilistic approaches such as [10,13,15] in that the focus is on the automatic generation of tolerance volume covering a given percentage.

An iterative scheme is developed in this paper to obtain a tolerance volume. The basic idea is as follows: (i) generate a sufficiently large tolerance volume and then (ii) reduce it until it covers V_T at the given confidence level. The iteration is based on the monotonic relation between the tolerance volume V_T and the confidence level of its coverage, which is proven in Section 3. (Intuitively, monotonicity is easy to see: as a tolerance volume shrinks from a 100% coverage, the confidence level of that volume also

decreases). Section 3 also details the ways of setting the initial tolerance volume and of reducing the volume.

Before the overall algorithm is presented, a procedure for computing the confidence level of covering a given tolerance volume is needed. For this computation, simulation or an analytic procedure can be used. Monte-Carlo simulation [2,4], while powerful, is computationally intensive. As the computation of the probability for coverage is an inner loop in the iterative scheme for generating the tolerance volume, speed is essential. An analytic procedure, which requires only O(n) computation under the given Jacobian matrix **J**, is given in Section 2 where n is the DOF. (It runs in less than 3 seconds on an IBM AT personal computer.) Finally, in Section 4, the run time of the iterative scheme (in less than 20 seconds on an IBM AT) is illustrated with a six-DOF Stanford manipulator.

2. Confidence Level of Tolerance Volume

A six-dimensional tolerance volume $\mathbf{V}_T$ is bounded by twelve hyperplanes: $-\Delta d_i^L \le \Delta d_i \le \Delta d_i^U$ for $1 \le i \le 6$. The probability of covering a given $\mathbf{V}_T$, called the *hit ratio HR*, is defined as:

$$HR = Pr\{\bigcap_{i=1}^{6} (-\Delta d_i^L \le \Delta d_i \le \Delta d_i^U)\} \tag{6}$$

To compute (6), the intersection of twelve hyperplane is integrated through the probability density function (p.d.f.) of the random variables Δd_i. However, the p.d.f. is multivariate and involves dependent variables. (The dependency can be seen from the example in Figure 2. In it, $\Delta x = -0.7071\,\Delta q_1 - 0.7071\,\Delta q_2$ and $\Delta y = 0.7071\,\Delta q_1 - 0.7071\,\Delta q_2$. Since Δx and Δy have the same random variables Δq_1 and Δq_2, Δx and Δy are not independent of each other.)

Using equation (1), the twelve hyperplanes can be rewritten as $G_k(\cdot) \ge 0$, for $1 \le k \le 12$, where

$$G_k(\Delta \mathbf{q}) = G_{2i-1}(\Delta \mathbf{q}) = \sum_{j=1}^{n} J_{ij}\Delta q_j + \Delta d_i^L \geq 0 \qquad \text{for k: odd and } 1 \leq i \leq 6$$

or $$G_k(\Delta \mathbf{q}) = G_{2i}(\Delta \mathbf{q}) = -\sum_{j=1}^{n} J_{ij}\Delta q_j + \Delta d_i^U \geq 0 \qquad \text{for k:even and } 1 \leq i \leq 6$$

Each of the twelve inequalities separate the $\Delta \mathbf{q}$ space into two half-spaces: *hit region* $\mathbf{H}_k$ and *miss region* $\mathbf{M}_k$, where

$$\mathbf{H}_k = \{ \Delta \mathbf{q} \mid G_k(\Delta \mathbf{q}) \geq 0 \} \quad \text{and}$$

$$\mathbf{M}_k = \{ \Delta \mathbf{q} \mid G_k(\Delta \mathbf{q}) < 0 \} .$$

Thus, the computation for the hit ratio in the $\Delta \mathbf{q}$ space becomes

$$HR = Pr\left(\bigcap_{k=1}^{12} \mathbf{H}_k\right) \tag{7}$$

While the computation of (7) still involves multiple integration, the p.d.f. for Δq_j is more manageable. The random variable Δq_j obeys the Central Limit Theorem [8] as the joint error comes from a large number of independent sources [4,15] such as inherent errors in the actuating motors, clearance between the gear teeth, encoder resolutions, computer round-offs, etc. In other words, Δq_j has a normal distribution with mean zero and standard deviation σ_j. Based on normality, a given limit of Δq_j, i.e., $\Delta q_j^L \leq \Delta q_j \leq \Delta q_j^U$, can be related to σ_j. For example, covering the limit of $\Delta q_j^L \leq \Delta q_j \leq \Delta q_j^U$ with a probability of 99.73% would yield

$$\sigma_j = \frac{\Delta q_j^L}{3} \qquad \text{for } 1 \leq j \leq n \tag{8}$$

where Δq_j^L is understood to be $\Delta q_j^L = \Delta q_j^U$ by symmetry.

The hit ratio (7) can then be computed by

$$HR = \int_{\bigcap_{k=1}^{12} \mathbf{H}_k} \cdots \int \phi_n(\mathbf{O};\mathbf{I}) \; d(\Delta q_1) \ldots d(\Delta q_n) \tag{9}$$

where $\phi_n(\mathbf{O};\mathbf{I})$ denotes the multivariate p.d.f. having the mean vector $\mathbf{O}$ and the identity correlation matrix $\mathbf{I}$.

2.1 Estimation of Hit Ratio

The intersection of the twelve hit regions, $\bigcap_{i=1}^{12} \mathbf{H}_k$, forms a convex polytope in the $\Delta\mathbf{q}$ space. To compute the probability of covering such a convex polytope under the multivariate normal distribution, techniques in numerical analysis or Monte-Carlo simulation may be used. While numerical integration yields an exact solution, it demands intensive computations. Simulation on the other hand offers a solution whose accuracy is proportional to the amount of computation time. Now, application to robotics demands speed and accuracy. To achieve both, a bounding formula attributed to Ditlevsen [7] in his work on reliability is adopted. This bounding formula uses only O(n) computations of univariate and bivariate normal distribution where n is the DOF.

Lemma 1: (Ditlevsen's Formula) HR is bounded by HR^L and HR^U, i.e.,

$$HR^L \le HR \le HR^U \tag{10}$$

where

$$HR^L = 1 - \sum_{k=1}^{12} Pr(\mathbf{M}_k) + \sum_{k=2}^{12} \max_{m<k} Pr(\mathbf{M}_k \cap \mathbf{M}_m)$$

$$HR^U = 1 - Pr(\mathbf{M}_1) - \sum_{k=2}^{12} \max \left\{ Pr(\mathbf{M}_k) - \sum_{m=1}^{k-1} Pr(\mathbf{M}_k \cap \mathbf{M}_m),\ 0 \right\}$$

[Proof] The proof can be found in [7].

The rest of this section is devoted to finding the probabilities $Pr(\mathbf{M}_k)$ and $Pr(\mathbf{M}_k \cap \mathbf{M}_s)$ for $1 \le k, s \le 12$, in the problem domain of kinematics generated by equation (1). To find $Pr(\mathbf{M}_k)$, first examine the distribution of $G_k(\Delta\mathbf{q})$ for $1 \le k \le 12$.

Lemma 2:

$$G_k(\Delta q) \sim \left\{ \begin{array}{ll} N\left(\Delta d_i^L, \sum_{j=1}^{n} J_{ij}^2 \sigma_j^2\right) & \text{if k : odd and i}=\frac{k+1}{2} \\ N\left(\Delta d_i^U, \sum_{j=1}^{n} J_{ij}^2 \sigma_j^2\right) & \text{if k : even and i}=\frac{k}{2} \end{array} \right\} \text{ for } 1 \le k \le 12$$

[Proof] See Appendix A.

Since $Pr(\mathbf{M}_k) = Pr(G_k(\Delta q)<0)$, Lemma 2 induces the following:

Lemma 3:

$$Pr(\mathbf{M}_k) = \left\{ \begin{array}{ll} \Phi\left(\dfrac{-\Delta d_i^L}{\sqrt{\sum_{j=1}^{n} J_{ij}^2\sigma_j^2}}\right) & \text{if k : odd and } i=\dfrac{k+1}{2} \\ \Phi\left(\dfrac{-\Delta d_i^U}{\sqrt{\sum_{j=1}^{n} J_{ij}^2\sigma_j^2}}\right) & \text{if k : even and } i=\dfrac{k}{2} \end{array} \right\} \quad \text{for } 1 \le k \le 12 \qquad (12)$$

where $\Phi(\cdot)$ is a cumulative standard normal distribution function.

[Proof] See Appendix B.

Now, Lemma 3 enables the looking up of $Pr(\mathbf{M}_k)$ from the standard normal distribution table. The remaining work for the computation of HR^L and HR^U is to find the joint probabilities $Pr(\mathbf{M}_k \cap \mathbf{M}_s)$ for $1 \le k,s \le 12$. First the correlations ρ_{ks} between the $G_k(\cdot)$'s are derived.

Lemma 4:

$$\rho_{ks} = \left\{ \begin{array}{ll} \mathbf{N}_k \cdot \mathbf{N}_s & \text{for } k<s^{\dagger\dagger} \\ \rho_{sk} & \text{for } k>s \\ 1 & \text{for } k=s \end{array} \right\} \qquad \text{for } 1 \le k,s \le 12 \qquad (13)$$

where $\cdot$ is a dot product of two $n \times 1$ vectors and $\mathbf{N}_k$ vector is defined as follows:

$$\mathbf{N}_k = \left\{ \begin{array}{ll} \left(\dfrac{J_{i1}\sigma_1}{\sqrt{\sum_{j=1}^{n} J_{ij}^2\sigma_j^2}}, \ldots, \dfrac{J_{in}\sigma_n}{\sqrt{\sum_{j=1}^{n} J_{ij}^2\sigma_j^2}}\right)^t & \text{if k:odd and } i=\dfrac{k+1}{2} \\ \left(\dfrac{-J_{i1}\sigma_1}{\sqrt{\sum_{j=1}^{n} J_{ij}^2\sigma_j^2}}, \ldots, \dfrac{-J_{in}\sigma_n}{\sqrt{\sum_{j=1}^{n} J_{ij}^2\sigma_j^2}}\right)^t & \text{if k:even and } i=\dfrac{k}{2} \end{array} \right\} \text{ for } 1 \le k \le 12 \qquad (14)$$

[Proof] See Appendix C.

---

†† In this case, provided additionally that s is an even number and s-k=1, then $\rho_{ks}=0$ since two hyperplanes $G_k(\Delta q)$ and $G_s(\Delta q)$ are parallel in Δq space.

Since it is known from Lemma 2 that $G_k(\Delta q)$ for $1 \le k \le 12$ follows the normal distribution and the correlation among them is computed by (13) and (14), the joint probabilities $Pr(\mathbf{M}_k \cap \mathbf{M}_s)$ can be obtained by using the bivariate normal p.d.f.

Lemma 5:

$$Pr(\mathbf{M}_k \cap \mathbf{M}_s) = Pr\{G_k(\Delta q) < 0, G_s(\Delta q) < 0\} \qquad 1 \le k, s \le 12 \qquad (15)$$

$$= \int_{-\infty}^{0}\int_{-\infty}^{0} \phi_{xy}(A, B, C, D, \rho_{ks})\, dx\, dy$$

where $\phi_{xy}(A, B, C, D, \rho_{ks})$ signifies a bivariate normal p.d.f. of the two random variables $x \sim N(A,B)$ and $y \sim N(C,D)$ having correlation ρ_{ks} and the determination of A,B,C, and D are given in Table 1.

		s	
		odd	even
k	odd	$A = \Delta d_g^L,\ B = \sum_{j=1}^{n} J_{gj}^2 \sigma_j^2$ $C = \Delta d_h^L,\ D = \sum_{j=1}^{n} J_{hj}^2 \sigma_j^2$ where $g = \frac{k+1}{2},\ h = \frac{s+1}{2}$	$A = \Delta d_g^L,\ B = \sum_{j=1}^{n} J_{gj}^2 \sigma_j^2$ $C = \Delta d_h^U,\ D = \sum_{j=1}^{n} J_{hj}^2 \sigma_j^2$ where $g = \frac{k+1}{2},\ h = \frac{s}{2}$
	even	$A = \Delta d_g^U,\ B = \sum_{j=1}^{n} J_{gj}^2 \sigma_j^2$ $C = \Delta d_h^L,\ D = \sum_{j=1}^{n} J_{hj}^2 \sigma_j^2$ where $g = \frac{k}{2},\ h = \frac{s+1}{2}$	$A = \Delta d_g^U,\ B = \sum_{j=1}^{n} J_{gj}^2 \sigma_j^2$ $C = \Delta d_h^U,\ D = \sum_{j=1}^{n} J_{hj}^2 \sigma_j^2$ where $g = \frac{k}{2},\ h = \frac{s}{2}$

Table 1. Parameters of the Bivariate Normal Distribution

2.2 Computing the Bounds of Hit Ratio

The flow chart for computing the lower and upper bounds of *HR* is summarized in Figure 4. The procedure consists of the three subroutines: the computation of hit ratio, the derivation of standard deviations, and the computation of the Jacobian matrix. Notice that the subroutine for computing the hit ratio needs twelve computations of the probability $Pr(\mathbf{M}_k)$ and $\binom{12}{2} = 66$ computations of the joint probabilities $Pr(\mathbf{M}_k \cap \mathbf{M}_s)$ $1 \le k,s \le 12$.

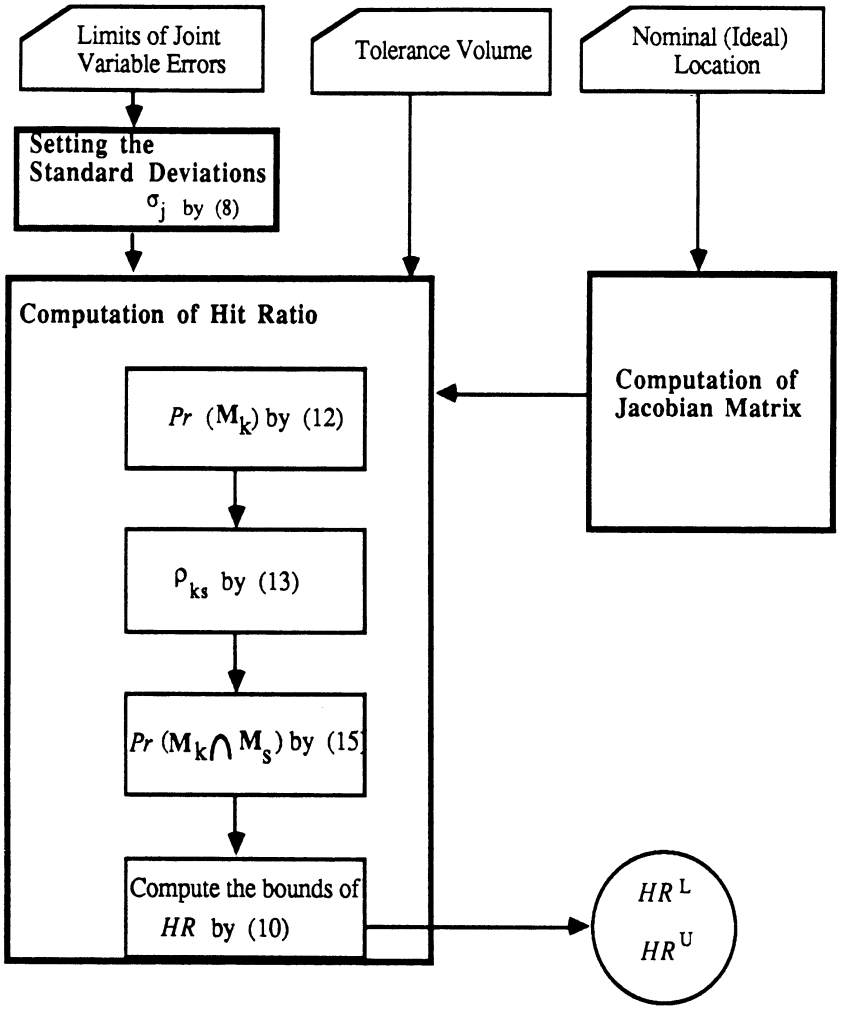


Figure 4. Flow-Chart for Computing the Lower and Upper Bounds of *HR*

In Figure 4, the tolerance volume as an input is determined by considering the characteristics of the operation to be done. For example, spot welding or painting is usually concerned with the positional tolerance of the end-effector across the work surface. In such a case, the orientation errors can be neglected by assigning artificially large values for the upper and lower limits when setting the tolerance volume. But, for operations requiring more delicate control of the six DOF, the specification of tolerance volume becomes quite complex since the specification of each DOF is interrelated with the specification of the other DOFs. In other words, for a complete specification of tolerance volume, the twelve interrelated limits of (6) (Δd_i^L and Δd_i^U for $1 \le i \le 6$) have to be decided simultaneously. To relieve this burden for input, the automatic generation of tolerance volume is considered in the next section using the probabilistic technique just presented.

3. Generation of Tolerance Volume

This section presents a procedure for generating a tolerance volume automatically, given the confidence level of that volume. The tolerance volume to be generated will be represented by a six-dimensional cuboid in $\Delta \mathbf{d}$ space such that $\Delta d_i^L \le \Delta d_i \le \Delta d_i^U$.

Among the twelve values, first consider only two, Δd_i^L and Δd_i^U, for an arbitrarily chosen i from $1 \le i \le 6$. If the confidence level of Δd_i is given as α_i, the problem is to determine the values Δd_i^L and Δd_i^U for ensuring

$$\int_{-\Delta d_i^L}^{\Delta d_i^U} \phi(\Delta d_i)\, d(\Delta d_i) = \alpha_i.$$

For uniqueness of the solution, it is assumed that $\Delta d_i^L = \Delta d_i^U$ (i.e., the confidence interval is symmetric around the mean value). This assumption is based on the fact that in the normal distribution the smallest

interval of covering a given confidence level is symmetric around the mean value. The value of $\Delta d_i^U (= \Delta d_i^L)$ can then be derived as follows:

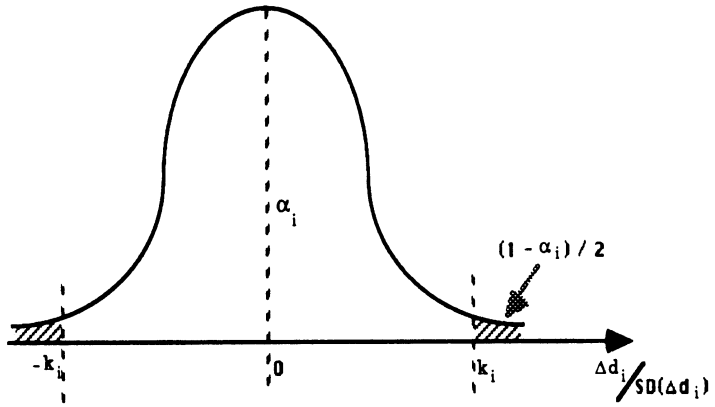
Lemma 6:

$$\Delta d_i^U = k_i \sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2} \qquad \text{for } 1 \le i \le 6$$

where the confidence coefficient k_i is equal to $\Phi^{-1}(1 - \frac{1-\alpha_i}{2})$.

[Proof] See Appendix D.

From Lemma 6, Δd_i^U can be determined from a given α_i. Hence, that which remains for the automatic generation of the twelve values of Δd_i^L and Δd_i^U (actually, six numbers under the assumption of symmetric confidence interval) is to determine the values of α_i $(1 \le i \le 6)$ from the desired confidence level α. To this end, the relationship between α and α_i is examined.



where $SD(\Delta d_i)$ is a standard deviation of Δd_i

Figure 5. Relationship between k_i and α_i

Lemma 7:

$$1 - \sum_{i=1}^{6} (1 - \alpha_i) \le \alpha \le \min(\alpha_1, \ldots, \alpha_6) \tag{16}$$

[Proof] See Appendix E.

Suppose that α is equally distributed into each DOF (i.e., $\alpha_1=\alpha_2= \ldots = \alpha_6$). Then, the inequality (16) can be converted to

$$6\alpha_i - 5 \leq \alpha \leq \alpha_i \tag{17}$$

From these inequalities of (17), the following can be observed: if α_i is assumed to be α for $1 \leq i \leq 6$, the hit ratio of the tolerance volume is then upper bounded by α_i; on the other hand, if α_i is determined by $\frac{\alpha + 5}{6}$, the hit ratio of the tolerance volume, α, is lower bounded by $6\alpha_i$-5. This observation implies that the desired α_i, which generates the tolerance volume of covering the given confidence level α, lies between the values of α and $\frac{\alpha+5}{6}$. That is, $\alpha_i^L \leq \alpha_i \leq \alpha_i^U$ where $\alpha_i^L=\alpha$ and $\alpha_i^U = \frac{\alpha+5}{6}$.

To find the desired α_i, bisection search is used with the two extreme values of α_i^L and α_i^U. Bisection is justified by the monotonic relations between α and Δd_i^U.

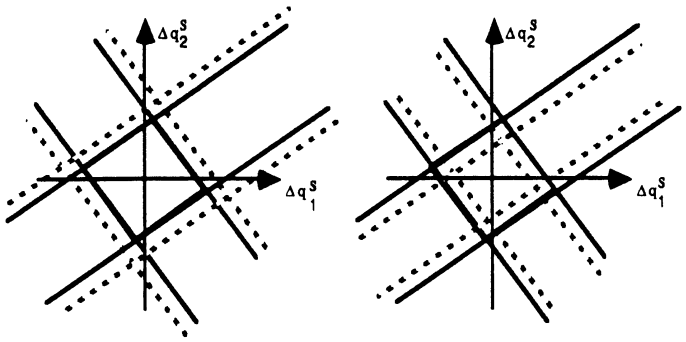
Lemma 8: As the confidence interval for each DOF increases (or decreases), the hit ratio of the tolerance volume also increases (or decreases). That is,

$$\frac{\partial \alpha}{\partial \Delta d_i^U} \geq 0 \qquad \text{for } 1 \leq i \leq 6$$

[Proof] See Appendix F.

For example, in a two DOF manipulator, the polytope expands (or shrinks) in a way as illustrated in Figure 6. This expansion (or shrinkage) of polytope holds for n DOF. Hence, the probability of covering the polytope, α, also increases (or decreases) with the increase (or decrease) of Δd_i^U.

The iterative scheme for generating the tolerance volume is summarized in Figure 7. Note that the computation of the hit ratio is an inner loop in step 5, which gives the two bounds, i.e., HR^L and HR^U. Fixing them to one value enables the comparison with α in step 6 . (HR^L is used in this paper such that the resultant tolerance volume is a little larger than the one covering at the exact confidence level α.)



(a) as Δd_i^U increases (b) as Δd_i^U decreases

Figure 6. Monotonic Relationship between Δd_i^U and α

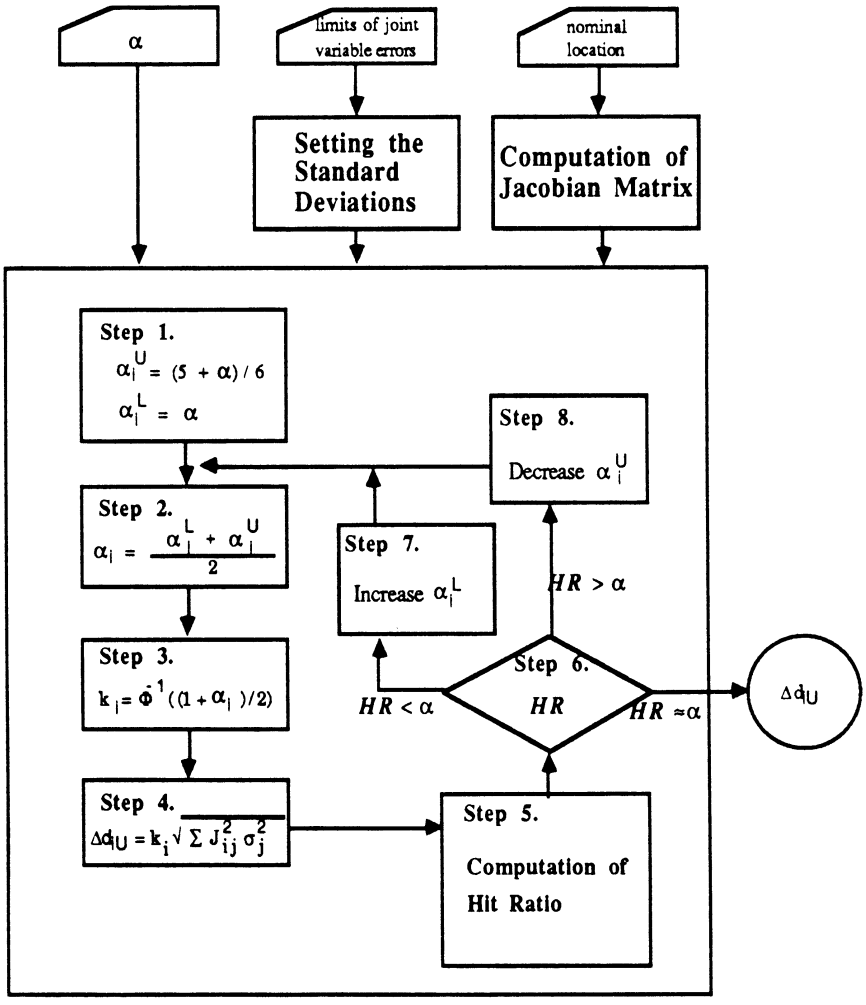


Figure 7. Flow Chart for Generating Tolerance Volume

4. Example

The iterative scheme for joint error analysis has been programmed in IBM turbo-PASCAL and runs on an IBM AT personal computer. Here, it is applied to the six DOF Stanford manipulator which consists of five revolute joints and one prismatic joint as shown in Figure 8. The kinematic parameters of the Stanford manipulator are listed in Table 2 based on the Denavit-Hartenberg notation [6], which is illustrated in Figure 9 for the adjacent joint coordinate frames. In Figure 9, θ_i, r_i, l_i, and a_i are the kinematic parameters for the joint angle, link offset, link length, and link twist, respectively, of the i-th joint. In Table 2, the joint variable q_3 for the third joint of the Stanford manipulator is the link offset r_3 since the third joint is prismatic. The other joint variables q_1, q_2, q_4, q_5, and q_6 are the joint angles θ_1, θ_2, θ_4, θ_5, and θ_6 since the corresponding joints are revolute.

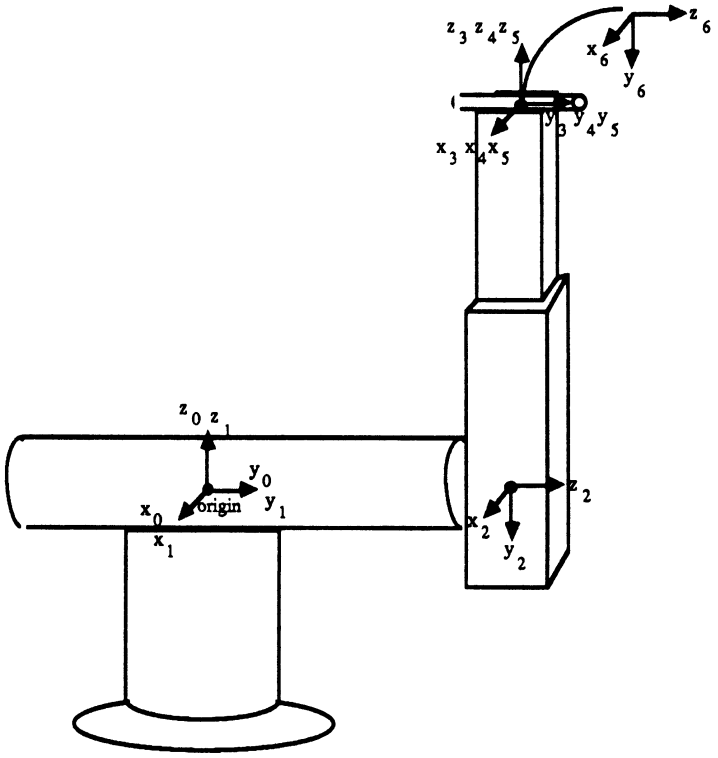


Figure 8. The Stanford Manipulator

Joint i	Kinematic Parameters				Joint Variable (q_i)
	a_i	l_i	r_i	θ_i	
1	-90^o	0	0	θ_1	θ_1
2	90^o	0	$r_2 = 20.0$	θ_2	θ_2
3	0^o	0	r_3	0	r_3
4	-90^o	0	0	θ_4	θ_4
5	90^o	0	0	θ_5	θ_5
6	0^o	0	0	θ_6	θ_6

Table 2. Kinematic Parameters of the Stanford Manipulator

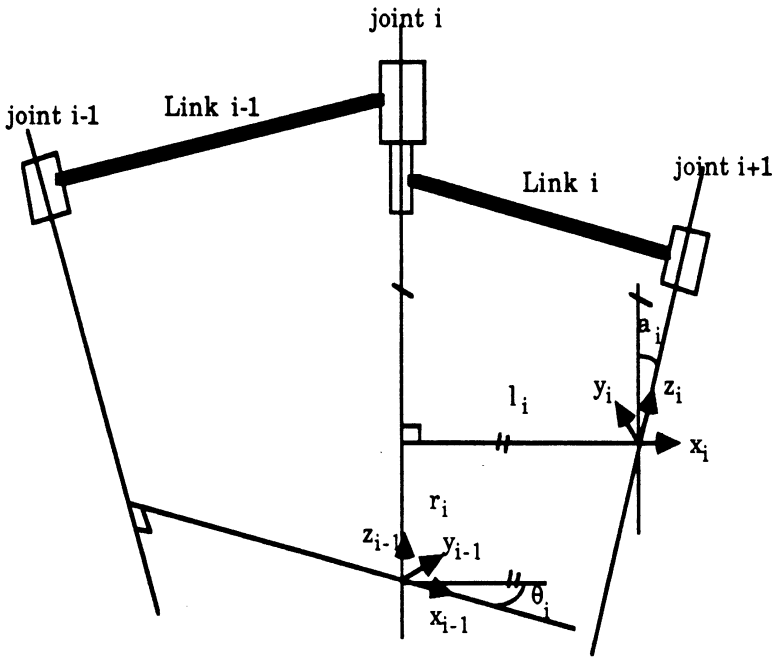


Figure 9. Denavit-Hartenberg Notation

Suppose that the second link offset, r_2, is set as 20.0 inches and that the errors of the joint variables are given as follows:

$$
\begin{aligned}
-1.0^{o} \le \Delta q_1 &\le 1.0^{o} \\
-1.0^{o} \le \Delta q_2 &\le 1.0^{o} \\
-1.0 \le \Delta q_3 &\le 1.0 \\
-0.5^{o} \le \Delta q_4 &\le 0.5^{o} \\
-0.5^{o} \le \Delta q_5 &\le 0.5^{o} \\
-0.5^{o} \le \Delta q_6 &\le 0.5^{o}
\end{aligned}
$$

Because of these joint errors, the actual location of the end-effector will deviate from the desired nominal location. One hundred nominal locations are randomly selected in the reachable workspace and tested. For each test, the deviation from the nominal location is computed as the tolerance volume according to the procedure in Figure 7. The tolerance volume covers at least 99.73%, i.e., α=0.9973. For comparison, the deterministic approach of [4] is also implemented. From the one hundred tests, the tolerance volume by the deterministic approach is on the average 5.04 times larger than the volume by the probabilistic approach. Table 3 shows a sample of the one hundred test points and their comparisons.

sample	1	2	3	...	27	28	...	84	85	...	100
Deterministic V_T^o	22.1	65.1	159.4	...	171.1	292.7	...	207.1	98.8	...	230.7
Probabilistic V_T^*	6.4	12.9	34.2	...	38.1	45.5	...	35.0	25.0	...	35.8
V_T^o / V_T^*	3.5	5.0	4.7	...	4.5	6.4	...	5.9	4.0	...	6.5

Note: Units of V_T^o and V_T^* are $(\text{inch})^3 * (\text{degree})^3$

Table 3. A Sample of One Hundred Test Points

To see the iterative scheme in detail, one specific test is taken here. The chosen nominal location is described by using the following transformation operators:

$$\text{Trans}(30.0, 6.0, 10.0)\cdot\text{Rot}((\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}), 45^{\circ}) \qquad (18)$$

where "Trans(·,·,·)" operator denotes a translation operation in x-, y-, and z-axis directions and "Rot(b,θ)" operator is a rotational operation performing a rotation about the unit axis direction **b** by an amount of θ degrees. Through inverse kinematics (the details of which is in [12]), the nominal location (18) can be executed by the following joint variables:

$$\mathbf{q}_N = (-29.51^o, 66.64^o, 25.22, 182.40^o, 30.26^o, 234.74^o)^t.$$

The Jacobian matrix **J** for this nominal location turned out to be the following matrix. (Again, the detailed procedure for computing Jacobian matrix can be found in [1,11,12]):

$$\begin{bmatrix} -6.000 & 8.702 & 0.799 & 0.000 & 0.000 & 0.000 \\ 30.000 & -4.926 & -0.452 & 0.000 & 0.000 & 0.000 \\ 0.000 & -23.152 & 0.397 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.493 & 0.000 & 0.799 & -0.478 & 0.506 \\ 0.000 & 0.870 & 0.000 & -0.452 & -0.878 & -0.311 \\ 1.000 & 0.000 & 0.000 & 0.397 & -0.038 & 0.805 \end{bmatrix}$$

This Jacobian matrix linearly transforms the given errors of the joint variables into a tolerance volume in the Cartesian space. With the normality of the joint variables around $\mathbf{q}_N$, the tolerance volume is obtained by taking the steps of Figure 7. The result is summarized in Figure 10. α_i is obtained by averaging the α_i^U and α_i^L according to step 2 in Figure 7. α_i is used to compute the confidence coefficient k_i in step 3. k_i defines the tolerance volume in step 4. The generated tolerance volume is then examined by computing the hit ratio *HR* in step 5 and by comparing *HR* with the desired confidence level of α in step 6. If *HR* is not sufficiently close to α, the value of α_i^U is then adjusted based on bisection in order to reassign α_i. The desired tolerance volume is obtained in six iterations and the steps consume 17.3 CPU seconds.

Each side of the generated tolerance volume corresponds to the confidence interval of each DOF in the Cartesian space. In the example, a confidence interval covers at least 99.95% ($=0.9973^{1/6}$). This probabilistic approach is compared with the deterministic approach in terms of the side length of the cubic tolerance volume. (Recall that the deterministic approach generates a volume V_T^o covering the actual tolerance volume V_T completely.) The comparison is rendered as Figure 11, in which the length of the six-sides is shown as a magnitude in the axis d_i^U, $1 \le i \le 6$. For example, for the first DOF the width of the

cuboid obtained by the deterministic approach is 1.107 times larger than that by the probabilistic approach. The entire tolerance volume by the deterministic approach V_T^o is 5.3 (= 1.107 x 1.310 x 1.216 x 1.645 x 1.436 x 1.271) times larger than V_T^*, the one by the probabilistic approach.

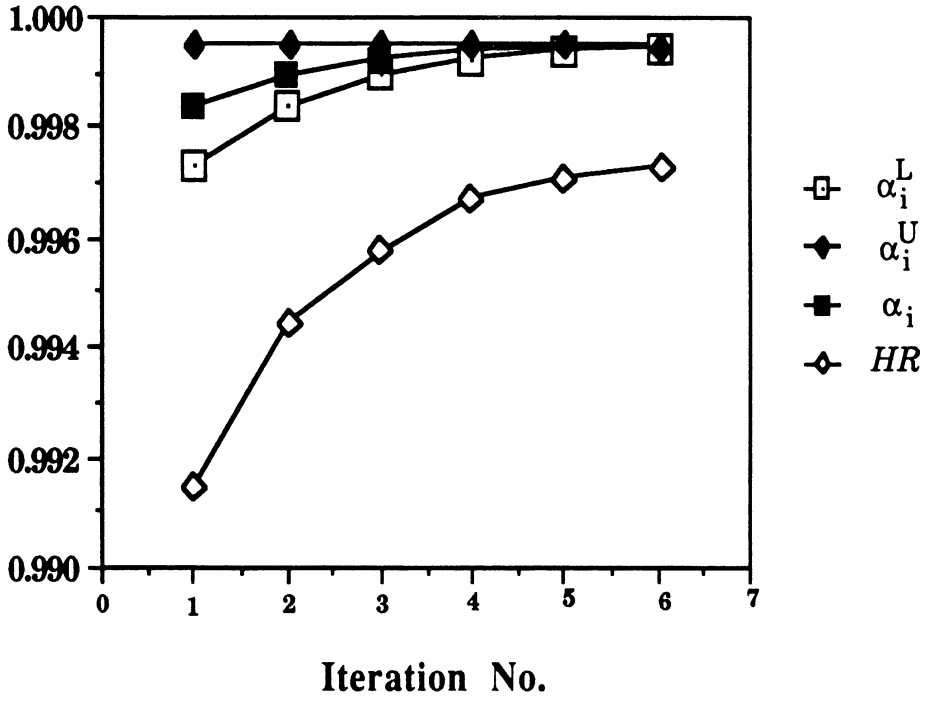


Figure 10. Iterations in the Joint Error Analysis

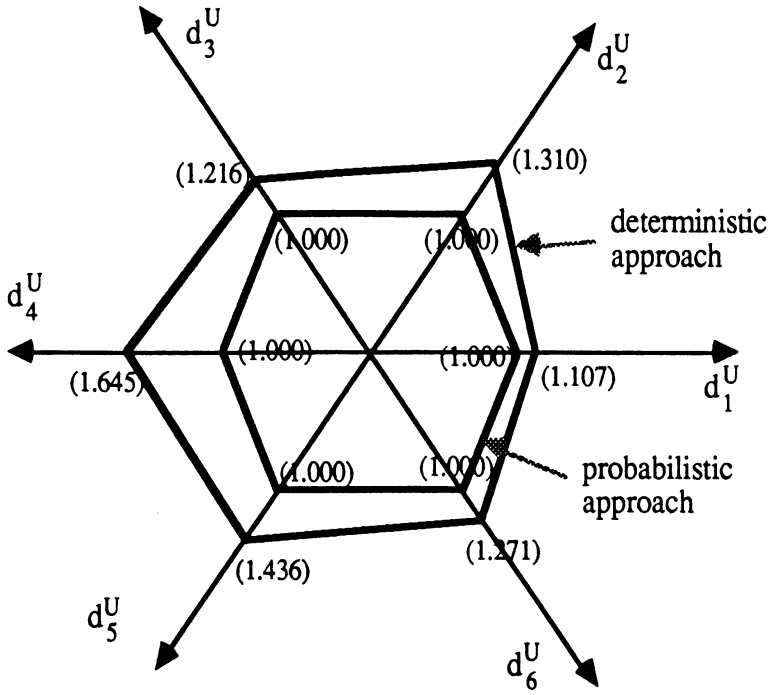


Figure 11. Probabilistic vs Deterministic Approaches

5. Conclusion

The ability of a robot to address a location repeatedly can now be evaluated accurately. Large improvement (of several hundred percent) over the worst case analysis is made possible by the probabilistic modeling of joint errors.

An n-dimensional tolerance volume, where n is the DOF of a manipulator, is computed not by simulation but analytically. The speed makes this analysis a useful tool.

Acknowledgement

Foundation grants from Ford Motor Company and Whirlpool Corporation have made this research possible.

References

[1] Asada, H. and Slotine J.-J. E., *Robot Analysis and Control*, John Wiely and Sons, Inc., New York, 1986.

[2] Azadivar, F., "The Effect of Joint Position Errors of Industrial Robots on Their Performance in Manufacturing Operations," *Int. J. Robotics Res.*, Vol. RA-3, No. 2, April 1987, pp. 109-114.

[3] Benhabib, B., Fenton, R.G., and Goldenberg, A.A., "Computer-Aided Joint Error Analysis of Robots," *IEEE J. of Robotics and Automation*, Vol. RA-3, No. 4, August 1987, pp. 317-322.

[4] Bhatti, P.K. and Rao, S.S., "Reliability Analysis of Robot Manipulator," in Advances in Design Automation - 1987: Volume 2 Robotics, Mechanisms, and Machine Systems, edited by Rao, S.S., *The 1987 ASME Design Technology Conferences - The Design Automation Conference*, Boston, MA, Sep. 27-30, 1987, pp. 45-53. (to appear in Journal of Mechanisms, Transmissions, and Automation in Design)

[5] Broderick, P.L. and Cipra, P.J., "A Method for Determining and Correcting Robot Position and Orientation Errors Due to Manufacturing," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 110, March 1988, pp. 3-10.

[6] Denavit, J. and Hartenberg, R.S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Trans. of ASME J. Applied Mechanics*, Jun. 1955, pp. 215-221.

[7] Ditlevsen, O.D., "Narrow Reliability Bounds for Structural Systems," *ASCE J. of Structural Mechanics*, Vol. 7, No. 4, Dec. 1979, pp. 453-472.

[8] Graybill, F.A., *An Introduction to Linear Statistical Model - Volume 1*, McGraw-Hill Book Co., Inc., New York, 1961.

[9] Hayati, S.A., "Robot Arm Geometric Link Parameter Estimation," *Proceedings of the 22nd IEEE Conf. on Decision and Control, San Antonio*, Texas, Dec. 1983, pp. 1477-1483.

[10] Kumar, K., "Positioning Accuracy of Manipulators with Encoder Equipped Joints," in *Robotics Research and Advanced Applications*, ASME pub., edited by W.J. Book, Nov., 1982, pp.35-42.

[11] Orin, D.E. and Schrader, W.W., "Efficient Computation of the Jacobian for Robot Manipulators," *Int. J. Robotics Res.*, Vol. 3, No. 4, Winter 1984, pp. 66-75.

[12] Paul, R.P., *Robot Manipulator: Mathematics, Programming, and Control*, Cambridge, MA, MIT Press, 1981.

[13] Waldron, K.J., "Positioning Accuracy of Manipulators," *Proceedings of the NSF Workshop on the Impact on the Academic Community of Required Research Activity for Generalized Robotic Manipulators*, Univ. of Florida, 1978.

[14] Whitney, D.E., Lozinski, C.A. and Rourke, J.M., "Industrial Robot Backward Calibration Method and Results," *Trans. of ASME, J. Dyn. Syst. Meas. Control*, Vol. 108, No. 1, 1986, pp. 1-8.

[15] Wu, C.H., "A Kinematic CAD Tool for the Design and Control of Robot Manipulator," *Int. J. Robotics Res.*, Vol. 3, No. 1, Spring 1984, pp. 58-67.

Appendices

Appendix A. Proof of Lemma 2.

$G_k(\Delta q)$ can be rewritten as

$$G_k(\Delta q) = \left\{ \begin{array}{ll} \Delta d_i + \Delta d_i^L & \text{if k : odd and i= } \frac{k+1}{2} \\ -\Delta d_i + \Delta d_i^U & \text{if k : even and i= } \frac{k}{2} \end{array} \right\} \quad \text{for } 1 \le k \le 12$$

Since Δd_i is a linear function of normal random variables Δq_j's, it also follows the normal distribution [8]. Its mean is then computed as the linear sum of the mean values of Δq_j's, and its variance is the linear square sum of the variances of Δq_j's; $\Delta d_i \sim N(0, \sum_{j=1}^{n} J_{ij}^2 \sigma_j^2)$. Hence, $G_k(\Delta q)$ follows the normal distribution as in (11) since Δd_i^L and Δd_i^U are given constants. Q.E.D.

Appendix B. Proof of Lemma 3.

For simplicity, assume that k is an odd number (as the case of even numbers can be done in the same way). Then,

$$Pr(M_k) = Pr(G_k(\Delta q)<0) = Pr\left(\frac{G_k(\Delta q) - \Delta d_i^L}{\sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2}} < \frac{-\Delta d_i^L}{\sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2}}\right).$$

Since $\dfrac{G_k(\Delta q) - \Delta d_i^L}{\sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2}} \sim N(0,1)$, the above probability is equal to $\Phi\left(\dfrac{-\Delta d_i^L}{\sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2}}\right)$. Q.E.D.

Appendix C. Proof of Lemma 4.

The correlation between two hyperplanes $G_k(\Delta q)$ and $G_s(\Delta q)$ is computed based on the Ditlevsen's observation [7]. In the standardized variable space it is equal to the dot product of the two outward unit normal vectors of the hyperplanes. To do so, first transform the normal random vector Δq into the standard

normal random vector Δq^* by the operation of $\Delta q_j^* = \frac{\Delta q_j}{\sigma_j}$ for $1 \le j \le n$. Then, the functions $G(\cdot)$ can be rewritten in terms of Δq^* as follows:

$$\left.\begin{aligned} G_{2i-1}(\Delta q) = G^*_{2i-1}(\Delta q^*) &= \sum_{j=1}^{n} J_{ij}\sigma_j \Delta q_j^* - \Delta d_i^L \\ G_{2i}(\Delta q) = G^*_{2i}(\Delta q^*) &= -\sum_{j=1}^{n} J_{ij}\sigma_j \Delta q_j^* + \Delta d_i^U \end{aligned}\right\} \text{ for } 1 \le i \le 6$$

From these equations, the outward normal vector of $G_k^*(\cdot)$ in Δq^* space, N_k for $1 \le k \le 12$, can be derived as shown in (14). Q.E.D.

Appendix D. Proof of Lemma 6.

As mentioned in the proof for Lemma 2, Δd_i follows the normal distribution $N(0, \sum_{j=1}^{n} J_{ij}^2\sigma_j^2)$. As illustrated in Figure 5, the confidence coefficient k_i is chosen such that $\Phi(k_i) = 1 - \frac{1-\alpha_i}{2}$. Then, Δd_i^U amounts to k times the standard deviation of Δd_i^U. Q.E.D.

Appendix E. Proof of Lemma 7.

The upper bound holds since the confidence level α of the convex polytope can not be greater than the individual confidence level α_i of the constituting half-spaces. The lower bound is based on the observation that the failure rate of the convex polytope, $1-\alpha$, is less than or equal to the sum of the individual failure rates $1-\alpha_i$. Q.E.D.

Appendix F. Proof of Lemma 8.

A geometric interpretation of the change of Δd_i^U is helpful here. Suppose the variable Δq_j is standardized through $\Delta q_j^S = \Delta q_j / \sigma_j$ such that $\Delta q_j^S \sim N(0,1)$ for $1 \le j \le n$. In the standardized space, the distance β_k from the origin to the transformed function is as follows:

$$\beta_k = \left\{ \begin{array}{ll} \dfrac{\Delta d_i^L}{\sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2}} & \text{for k:odd and } i = \dfrac{k+1}{2} \\ \dfrac{\Delta d_i^U}{\sqrt{\sum_{j=1}^{n} J_{ij}^2 \sigma_j^2}} & \text{for k:even and } i = \dfrac{k}{2} \end{array} \right\} \quad \text{for } 1 \le k \le 12 \qquad (19)$$

From (19), it becomes clear that the increase (or decrease) of Δd_i^U makes the corresponding distance β_k greater (or smaller). Therefore $\dfrac{\partial \alpha}{\partial \Delta d_i^U} \ge 0$ since $\dfrac{\partial \alpha}{\partial \beta_k} \ge 0$. Q.E.D.