Chapter 3: Inverse Kinematics by Local Optimization

This chapter discusses several methods of solving the inverse kinematics problem using local optimization. The methods fall into two categories: analytical and direct search. The analytical methods optimize a single performance criterion while simultaneously solving the inverse kinematics problem for a redundant robot. The direct search method of solving the inverse kinematics problem is one of the major developments in this dissertation. This chapter develops the direct search method for robots with six degrees of freedom. The next chapter extends the method to include redundant robots.

Analytical methods can solve the inverse kinematics problem at the position, velocity, and acceleration levels. At the position level, the problem is highly non-linear and closed-form solutions only exist for special cases. Due to the instantaneous nature of the relationships, the problem is linear at the velocity and acceleration levels. This chapter derives analytical solutions minimizing joint speeds and kinetic energy at the velocity level. At the acceleration level, solutions minimizing joint accelerations and joint torques are derived.

In the most basic sense, direct search describes a method of solving problems numerically using sets of trial solutions to guide a search. The search is direct because it does not explicitly evaluate derivatives, as do gradient and higher-order numerical methods. Because the direct search does not use derivatives, it is stable at singularities. Direct search's name may give the impression that it is somehow less analytical than other numerical methods. This

is not true. For all but the simplest of problems, all numerical methods are, in essence, searches. This is the nature of numerical methods. Most of them, including Newton-Raphson and predictor-corrector, have an analytical basis for linear or quadratic functions. For functions of degree higher than two, the analytical basis disappears. As the non-linearity increases, the relationship between a function's derivatives and its solution becomes less and less clear. Nothing exemplifies this more than a robot at a singularity. The derivative in at least one direction is undefined, yet there is still a solution. A sixteenth degree equation describes the inverse kinematic relationship for a general robot with six degrees of freedom (Duffy, 1980). For this degree of non-linearity, all numerical methods are searches.

Most direct search strategies share an amount of commonalty. The search begins and an estimate of the solution, which serves as an initial base point. A pattern of local explorations around the base point generates information about the locality. A decision making strategy chooses a new base point based on this information and then the search proceeds with another series of explorations. This process continues until the search, at least a successful one, finds a solution. This chapter describes three different exploration patterns and three different decision making strategies. These exploration patterns and decision making strategies combine to form a number of different direct methods. Some of these methods guarantee optimality and some do not. This chapter discusses the reliability of the various methods. Finally, the chapter concludes with a discussion as to each of the method's relative computational complexity and their recommended applications.

3.1 ANALYTICAL SOLUTIONS

An analytical solution to the inverse kinematics problem that simultaneously optimizes a performance criterion can be formulated at the position, velocity and acceleration levels. At the position level, the non-linear mapping between joint space and end-effector space is

$$\mathbf{x} = \mathbf{f}(\Phi) \tag{3.1}$$

where the vector $\mathbf{x}$ represents the position and orientation of the end-effector, the vector Φ represents the joint displacements, and $\mathbf{f}(\Phi)$ is a description of the robot's input-output positional relationships.

Differentiating Equation (3.1) relates the joint speeds to the end-effector velocity.

$$\dot{\mathbf{x}} = \mathbf{J}(\Phi)\dot{\Phi} \tag{3.2}$$

Differentiating Equation (3.2) relates the end-effector acceleration to the Jacobian, $\mathbf{J}(\Phi)$, the time rate of change of the Jacobian, $\dot{\mathbf{J}}(\Phi)$, the joint speeds, $\dot{\Phi}$, and the joint accelerations, Φ.

$$\ddot{\mathbf{x}} = \mathbf{J}(\Phi)\ddot{\Phi} + \dot{\mathbf{J}}(\Phi)\dot{\Phi} \tag{3.3}$$

Because Equations (3.2) and (3.3) provide an instantaneous relationship between joint states and end-effector states, they are inherently linear in terms of the higher order properties. This linearity provides a unique relationship for the joint speeds or joint accelerations much simpler than solutions for the joint displacements. Integrating the joint speeds once, or the accelerations twice, gives the joint displacements that represent solutions to the inverse kinematics problem.

3.1.1 Position-level Optimization

At the position level, the inverse kinematics problem for a redundant robot is very complex. It may be posed as an optimization problem with very coupled and non-linear constraint equations. Only extremely simple sub problems, such as the planar case, have closed-form solutions. A general formulation for the optimization of a single position-level criterion is obtained using the classical LaGrange multiplier technique.

To optimize a position level criterion of the form

$$c(\Phi) \tag{3.4}$$

begin by rewriting Equation (3.1) as an equality constraint.

$$\mathbf{x} - \mathbf{f}(\Phi) = \mathbf{0} \tag{3.5}$$

Adjoining the constraint set, Equation (3.5), to the performance criterion, Equation (3.4), via a LaGrange multiplier vector, λ, forms the Hamiltonian

$$H(\Phi, \lambda) = c(\Phi) + \lambda^{\mathrm{T}}[\mathbf{x} - \mathbf{f}(\Phi)] \tag{3.6}$$

The conditions for optimality are

$$\frac{\partial H}{\partial \lambda} = \mathbf{0} \tag{3.7}$$

and

$$\frac{\partial H}{\partial \Phi} = \mathbf{0} \tag{3.8}$$

The exact form of the equations described by (3.7) and (3.8) depends upon the robot geometry as well as the performance criterion. Not only must Φ be found, as in the fully-constrained case, but the LaGrange multiplier, λ, must be found as well.

3.1.2 Minimizing Joint Speeds

The inverse kinematics problem with optimization is linear at the velocity level. It is linear because of the instantaneous nature of the velocity-level equations. The LaGrange multiplier technique solves the problem in a straightforward fashion. A general solution minimizing the sum of the squares of the joint speeds is developed below.

A measure of the magnitude of the joint speeds is written as

$$c(\Phi) = \dot{\Phi}^{T}\dot{\Phi} \tag{3.9}$$

A solution which minimizes this performance criterion while simultaneously finding the required joint speeds is readily formulated. Equation (3.2) is rewritten as an equality constraint

$$\dot{\mathbf{x}} - \mathbf{J}\dot{\Phi} = \mathbf{0} \tag{3.10}$$

Adjoining the constraint set, Equation (3.9), to the performance criterion via a LaGrange multiplier forms the Hamiltonian.

$$H\left(\dot{\Phi}, \lambda\right) = \dot{\Phi}^{T}\dot{\Phi} + \lambda^{T}\left[\dot{\mathbf{x}} - \mathbf{J}\dot{\Phi}\right] \tag{3.11}$$

The conditions for optimality are

$$\frac{\partial H}{\partial \lambda} = \mathbf{0} \tag{3.7}$$

and

$$\frac{\partial H}{\partial \dot{\Phi}} = \mathbf{0} \tag{3.12}$$

Rewriting Equation (3.11) by taking the partial derivative with respect to $\dot{\Phi}$ gives

$$2\dot{\Phi}^{T} - \lambda^{T}\mathbf{J} = 0 \tag{3.13}$$

Solving Equation (3.13) for $\dot{\Phi}$ gives

$$\dot{\Phi} = \frac{1}{2}\mathbf{J}^{\mathrm{T}}\lambda \tag{3.14}$$

Now, rewriting Equation (3.11) by taking the partial derivative with respect to λ gives

$$\dot{\mathbf{x}} - \mathbf{J}\dot{\Phi} = 0 \tag{3.15}$$

and substituting for $\dot{\Phi}$ as given by Equation (3.14) gives

$$\dot{\mathbf{x}} - \frac{1}{2}\mathbf{J}\mathbf{J}^{\mathrm{T}}\lambda = 0 \tag{3.16}$$

It is now possible to solve for the LaGrange multipliers as

$$\lambda = 2\left[\mathbf{J}\mathbf{J}^{\mathrm{T}}\right]^{-1}\dot{\mathbf{x}} \tag{3.17}$$

This step requires the crucial assumption that the Jacobian is full-rank. Given that the Jacobian is full-rank, $\mathbf{J}\mathbf{J}^{\mathrm{T}}$ is a full-rank and square matrix which can be uniquely inverted as indicated in Equation (3.17). The Jacobian is not full-rank at singularities, where this method fails. Assuming, however, that the Jacobian is full-rank and substituting for λ in Equation (3.16) gives

$$\dot{\Phi} = \mathbf{J}^{\mathrm{T}}\left[\mathbf{J}\mathbf{J}^{\mathrm{T}}\right]^{-1}\dot{\mathbf{x}} \tag{3.18}$$

The general form of this equation is well-known and seductively simple. Several comments are in order. First, Equation (3.18) only optimizes a single criterion. That criterion is extremely simple and only loosely associated with any improvement in robot performance. Second, this solution fails at singularities. Finally, the solution does not consider position-level constraints, such as joint limits. Taken together, these characteristics make Equation (3.18) unsuitable for practical application. The derivation is, however, correct mathematically. Thus,

Equation (3.18) is a general solution to the inverse kinematics problem that minimizes the sum of the squares of the joint speeds.

3.1.2 Minimizing Kinetic Energy

The kinetic energy content of the robot is a performance criterion with a more subtle physical interpretation. Minimizing this criterion may give increased precision and higher operating speeds. The kinetic energy criterion is

$$c\left(\dot{\Phi}\right) = \dot{\Phi}^{\mathrm{T}}\left[\mathbf{I}^{*}\right]\dot{\Phi} \tag{3.19}$$

where $\mathbf{I}^{*}$ is the inertia tensor for the robot. A solution minimizing this performance criterion while simultaneously solving for the required joint speeds may be derived. Equation (3.2) is rewritten as an equality constraint

$$\dot{\mathbf{x}} - \mathbf{J}\dot{\Phi} = \mathbf{0} \tag{3.20}$$

Adjoining the constraint set, Equation (3.20), to the performance criterion via a LaGrange multiplier vector forms the Hamiltonian.

$$H\left(\dot{\Phi}, \lambda\right) = \dot{\Phi}^{\mathrm{T}}\left[\mathbf{I}^{*}\right]\dot{\Phi} + \lambda^{\mathrm{T}}\left[\dot{\mathbf{x}} - \mathbf{J}\dot{\Phi}\right] \tag{3.21}$$

The conditions for optimality are

$$\frac{\partial H}{\partial \lambda} = \mathbf{0} \tag{3.7}$$

and

$$\frac{\partial H}{\partial \dot{\Phi}} = \mathbf{0} \tag{3.12}$$

Rewriting Equation (3.21) by taking the partial derivative with respect to $\dot{\Phi}$ gives

$$2\dot{\Phi}^{\mathrm{T}}\mathbf{I}^{*} - \lambda^{\mathrm{T}}[\mathbf{J}] = 0 \tag{3.22}$$

and solving Equation (3.22) for Φ gives

$$\dot{\Phi} = \frac{1}{2}\left[\mathbf{I}^*\right]^{-1}\mathbf{J}^T\lambda \tag{3.23}$$

Now, rewriting Equation (3.21) by taking the partial derivative with respect to λ gives

$$\dot{\mathbf{x}} - \mathbf{J}\dot{\Phi} = 0 \tag{3.24}$$

Substituting for $\dot{\Phi}$ as given by Equation (3.23) gives

$$\dot{\mathbf{x}} - \frac{\mathbf{1}}{\mathbf{2}}\mathbf{J}\left[\mathbf{I}^*\right]^{-1}\mathbf{J}^T\lambda = 0 \tag{3.25}$$

The LaGrange multipliers are now found as

$$\lambda = 2\left[\mathbf{J}\left[\mathbf{I}^*\right]^{-1}\mathbf{J}^T\right]^{-1}\dot{\mathbf{x}} \tag{3.26}$$

Again this step requires a full-rank Jacobian. It also requires that $\mathbf{I}^*$ is definite (which it is). Given these assumptions, $\mathbf{J}\left[\mathbf{I}^*\right]^{-1}\mathbf{J}^T$ is a full-rank square matrix and can be uniquely inverted. Substituting for λ in Equation (3. 23) gives

$$\dot{\Phi} = \left[\mathbf{I}^*\right]^{-1}\mathbf{J}^T\left[\mathbf{J}\left[\mathbf{I}^*\right]^{-1}\mathbf{J}^T\right]^{-1}\dot{\mathbf{x}} \tag{3.27}$$

This equation provides the joint speeds .

Again, this derivation is only good when the Jacobian is full-rank. By minimizing system kinetic energy, Equation (3.27) may enhance performance under certain conditions. Equation (3.27) does not, however, consider position-level constraints which may make it unsuitable for practical applications. Nonetheless, this solution is general in the sense that it finds the minimum kinetic energy solution for any serial robot geometry.

3.1.2 Minimizing Joint Accelerations

It is also possible to formulate the inverse kinematics problem with optimization at the acceleration level. As with the velocity-level solutions, these relationships are linear and are thus much easier to solve than the position-level equations. General solutions for minimizing the joint accelerations and the joint torques may be derived.

A performance criterion which measures the magnitude of the change in joint speeds with respect to time is

$$c\left(\ddot{\Phi}\right) = \ddot{\Phi}^{\mathrm{T}}\ddot{\Phi} \tag{3.28}$$

The solution which minimizes this criterion while simultaneously solving for the required joint accelerations may be found using the method of LaGrange multipliers. Equation (3.3) is rewritten as an equality constraint

$$\ddot{\mathbf{x}} - \mathbf{J}\ddot{\Phi} - \dot{\mathbf{J}}\dot{\Phi} = \mathbf{0} \tag{3.29}$$

Adjoining the constraint set, Equation (3.29), to the performance criterion, Equation (3.28), via a LaGrange multiplier forms the Hamiltonian.

$$H\left(\ddot{\Phi}, \lambda\right) = \ddot{\Phi}^{\mathrm{T}}\ddot{\Phi} + \lambda^{\mathrm{T}}\left(\ddot{\mathbf{x}} - \mathbf{J}\ddot{\Phi} - \dot{\mathbf{J}}\dot{\Phi}\right) \tag{3.30}$$

The conditions for optimality are

$$\frac{\partial H}{\partial \lambda} = \mathbf{0} \tag{3.7}$$

and

$$\frac{\partial H}{\partial \ddot{\Phi}} = \mathbf{0} \tag{3.31}$$

Rewriting Equation (3.30) by taking the partial derivative with respect to $\ddot{\Phi}$ gives

$$2\ddot{\Phi}^T - \lambda^T \mathbf{J} = 0 \tag{3.32}$$

Solving Equation (3.32) for $\ddot{\Phi}$ gives

$$\ddot{\Phi} = \frac{1}{2}\mathbf{J}^T\lambda \tag{3.33}$$

Now, rewriting Equation (3.30) by taking the partial derivative with respect to λ gives

$$\ddot{\mathbf{x}} - \mathbf{J}\ddot{\Phi} - \dot{\mathbf{J}}\dot{\Phi} = \mathbf{0} \tag{3.34}$$

and substituting for $\ddot{\Phi}$ as given by Equation (3.33) gives

$$\ddot{\mathbf{x}} - \frac{1}{2}\mathbf{J}\mathbf{J}^T\lambda - \dot{\mathbf{J}}\dot{\Phi} = 0 \tag{3.35}$$

The LaGrange multipliers may now be found as

$$\lambda = 2\left[\mathbf{J}\mathbf{J}^T\right]^{-1}\left[\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\Phi}\right] \tag{3.36}$$

Which assumes the Jacobian is full-rank. Substituting for λ in Equation (3.35) gives

$$\ddot{\Phi} = \mathbf{J}^T\left[\mathbf{J}\mathbf{J}^T\right]^{-1}\left[\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\Phi}\right] \tag{3.37}$$

Equation (3.37) represents a solution to the inverse kinematics problem which simultaneously minimizes the sum of the squares of the joint accelerations.

As expected, Equation (3.37) is only valid if the Jacobian is full-rank.

3.1.1 Minimization of Joint Torques

It is also possible to solve for the joint torques which, neglecting modeling errors, result in the desired end-effector motion. The sum of the squares of the joint torques is the performance criterion.

$$c(\tau) = \frac{1}{2}\tau^{\mathrm{T}}\tau \tag{3.38}$$

The vector τ represents the joint torques. The joint torques are given by the relationship

$$\tau = \mathbf{I}^{*}\ddot{\Phi} + \mathbf{c} + \mathbf{g} \tag{3.39}$$

where $\mathbf{I}^{*}$ represents the inertia tensor, $\mathbf{c}$ represents the centripetal and Coriolis acceleration terms, and $\mathbf{g}$ the effect of gravity. Writing the joint accelerations in terms of the joint torques gives

$$\ddot{\Phi} = \left[\mathbf{I}^{*}\right]^{-1}(\tau - \mathbf{c} - \mathbf{g}) \tag{3.40}$$

Substituting the joint accelerations into the original acceleration-level constraints on the end-effector motion, Equation (3.3), gives an equality constraint in terms of the joint torques

$$\ddot{\mathbf{x}} - \mathbf{J}\left[\mathbf{I}^{*}\right]^{-1}(\tau - \mathbf{c} - \mathbf{g}) - \dot{\mathbf{J}}\dot{\Phi} = \mathbf{0} \tag{3.41}$$

Adjoining the constraint set, Equation (3.41), to the performance criterion, Equation (3.38), forms the Hamiltonian.

$$H(\tau, \lambda) = \frac{1}{2}\tau^{\mathrm{T}}\tau + \lambda^{\mathrm{T}}\left[\ddot{\mathbf{x}} - \mathbf{J}\left[\mathbf{I}^{*}\right]^{-1}(\tau - \mathbf{c} - \mathbf{g}) - \dot{\mathbf{J}}\dot{\Phi}\right] \tag{3.42}$$

The conditions for optimality are

$$\frac{\partial H}{\partial \lambda} = \mathbf{0} \tag{3.7}$$

and

$$\frac{\partial H}{\partial \tau} = \mathbf{0} \tag{3.43}$$

Taking the partial derivatives as indicated by Equation (3.7) gives

$$\ddot{\mathbf{x}} - \mathbf{J}\left[\mathbf{I}^*\right]^{-1}(\tau - \mathbf{c} - \mathbf{g}) - \dot{\mathbf{J}}\dot{\Phi} = \mathbf{0} \tag{3.44}$$

Taking the partial derivatives as indicated by Equation (3.43) gives

$$\tau - \left[\mathbf{I}^*\right]^{-1}\mathbf{J}^{\mathrm{T}}\lambda = \mathbf{0} \tag{3.45}$$

Solving Equations (3.44) and (3.45) for τ and equating the result gives the LaGrange multiplier, λ. Substituting the LaGrange multiplier into (3.45) gives

$$\tau = \left[\mathbf{I}^*\right]^{-\mathrm{T}}\mathbf{J}^{\mathrm{T}}\left[\mathbf{J}\left[\mathbf{I}^*\right]^{-1}\left[\mathbf{I}^*\right]^{-\mathrm{T}}\mathbf{J}^{\mathrm{T}}\right]^{-1}\left[\ddot{\mathbf{x}} - \mathbf{J}\left[\mathbf{I}^*\right]^{-1}(\mathbf{c}+\mathbf{g}) - \dot{\mathbf{J}}\dot{\Phi}\right] \tag{3.46}$$

Equation (3.46) minimizes the sum of the squares of the joint torques while simultaneously producing the desired end-effector motion. Other researchers have also derived this solution (Carnigan, 1991). In a practical application of Equation (3.46), the accuracy of the end-effector motion is determined by the accuracy of the dynamic model represented by Equation (3.39). Since no model is perfect, some sort of feedback to correct end-effector errors would have to be included if this method were applied to an actual robot.

3.2 Direct Search

Each of the preceding examples expressed the inverse kinematics problem as an unconstrained minimization problem using LaGrange multipliers. Calculus was then used for the actual minimization. The numerical solution of the inverse kinematics problem may employ an analogous technique. The displacement constraints on the end-effector are expressed as an unconstrained error function. For instance, the Cartesian error between the current and the desired displacement of the end-effector is one possible unconstrained error function. The error function is minimized using numerical minimization techniques. A solution to the

inverse kinematics problem is found if the error is minimized to zero. Direct search is one numerical minimization technique applied to the inverse kinematics problem in this manner.

At its most basic level, direct search is a method of solving problems numerically using sets of trial solutions to guide a search. The search is successful if an actual solution is found. Direct search methods use only function values and are distinguished from gradient and higher-order methods requiring values for the function's derivatives. Because a direct search does not require a function's derivatives, an inverse kinematics implementation based on direct search may converge at singularities.

Most direct search strategies share a degree of commonalty. The search begins at an estimate of the solution. This estimate serves as an initial base point for the search. A pattern of explorations about the base point generates a set of trial solutions. The trial solutions are evaluated and a decision making process selects one trial solution as the next base point. Another pattern of local exploration is then performed about this new base point. This process of local exploration and decision making continues until the search, at least a successful one, finds a solution. Different exploration patterns and different decision making processes may be combined to form a number of direct search strategies. The rest of this chapter develops direct search inverse kinematics techniques for robots with six degrees of freedom. Chapter 4 extends direct search to the multicriteria inverse kinematics problem for redundant robots.

3.2.1 Transformation of End-effector Constraints

The inverse kinematics problem may be expressed as an unconstrained optimization problem by transforming the end-effector constraints via an error function. The error represents the difference between the desired and the current end-effector displacements. In a direct search, the error is calculated for each of the trial solutions. Much of the generality of direct search as an inverse kinematics method is associated with expressing the end-effector constraints as a transformed error function. The error is found using the forward transformation equations. The forward transformation equations determine the position and orientation of the end-effector for a given set of joint displacements. The Denavit and Hartenberg kinematic parameters for the robot determine the forward transformation equations in an entirely general fashion.

Cartesian coordinates (in translation and rotation) are a natural choice for expressing the error function. There are several ways of writing the error in terms of Cartesian coordinates. One of them is as the sum of the errors in position and orientation.

$$E = E_p + cE_O \tag{3.47}$$

In Equation (3.47), E represents the total error, E_P the error in position, E_O the error in orientation, and c a scaling parameter to balance the units and magnitudes between position and orientation. The error in position is calculated as

$$E_P = \sqrt{(x_T - x_D)^2 + (y_T - y_D)^2 + (z_T - z_D)^2} \tag{3.48}$$

where x y, and z represent the translations. The subscript T indicates trial solution and the subscript D indicates desired solution. The error in orientation is

calculated in a similar manner using the three end-effector orientations rather than the translations. *XYZ* fixed angles are used here, but there are many ways of describing the end-effector orientation and any of them may be used in a similar fashion. *XYZ* fixed angles are three successive rotations about the X, Y, and Z axes of a fixed reference frame.

$$E_O = \sqrt{(R_{xT} - R_{xD})^2 + (R_{yT} - R_{yD})^2 + (R_{zT} - R_{zD})^2} \tag{3.49}$$

In this equation, R represents the magnitude of the rotation. The subscripts x, y, and z represent the axes of rotation. The subscript T indicates trial solution, and the subscript D indicates desired solution.

The scaling parameter in Equation (3.47), c, may be chosen in a variety of ways. Chapter 4 discusses procedures for calculating scaling parameters in detail. Providing comparable magnitudes of error in position and orientation is the basic goal when calculating the scaling parameter.

Calculating the error in the manner of Equations (3.48) and (3.49) requires calculating the end-effector placement corresponding to a given trial solution. Calculating the end-effector placement involves the forward transformation equations. The forward transformation equations may be written for a general robot geometry in terms of Denavit and Hartenberg parameters. These parameters assign a local reference frame to each of the robot's joint axes.

For a given robot, a local reference frame, {i}, is related to the previous local frame, {i - 1}, by two screw operators

$${}^{i-1}_{i}\mathbf{T} = \text{Screw}_x(a_{i-1}, \alpha_{i-1})\text{Screw}_z(d_i, \Phi_i) \tag{3.50}$$

where ${}^{i-1}_{i}\mathbf{T}$ denotes the transform relating the two frames. $\text{Screw}_x(a_{i-1}, \alpha_{i-1})$ denotes a translation along the x-axis by the link length a_{i-1} and a rotation about the x-axis by the twist angle α_{i-1}. Similarly, $\text{Screw}_z(d_i, \Phi_i)$ denotes a translation along the z-axis by the offset d_i and a rotation about the z-axis by the joint variable Φ_i. These two screw operations completely define the transform relating frame {i} to frame {i - 1}.

$$ {}^{i-1}_{i}\mathbf{T} = \begin{vmatrix} \cos\Phi_i & -\sin\Phi_i & 0 & a_{i-1} \\ \sin\Phi_i\cos\alpha_{i-1} & \cos\Phi_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\Phi_i\sin\alpha_{i-1} & \cos\Phi_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{3.51} $$

This equation represents a homogeneous transform because it contains both rotational and translational elements. In a general fashion, concatenating these transforms produces a single transform which relates the final frame {n} to the base frame {0}.

$$ {}^{0}_{n}\mathbf{T} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T} \ldots {}^{n-1}_{n}\mathbf{T} \tag{3.52} $$

The homogeneous transform ${}^{0}_{n}\mathbf{T}$ represents the kinematic displacement equations in joint space and completely defines the end-effector placement for a given trial solution.

Extracting the end-effector placement from the homogeneous transform ${}^{0}_{n}\mathbf{T}$ forms an expression for the Cartesian error as given in Equation (3.47). The translational portion appears explicitly as the first three numbers in the last column of ${}^{0}_{n}\mathbf{T}$ and thus presents no difficulty. There is, however, a singularity associated with extracting the rotational portion. This singularity depends on the

particular angle set convention describing the orientation of the end-effector. In the case of *XYZ* fixed rotation angles, there is a singularity when the rotation about the *Y* axis is ninety degrees. In this case, the *X* and *Z* axes of rotation are collinear and only their sum may be computed. Logic must be included to detect and resolve the singularity. Including only the sum of the *X* and *Z* rotations when formulating the rotational portion of the Cartesian error function at this singularity is a simple resolution.

$$E_O = \sqrt{\left[\left(R_{xT} + R_{zT}\right) - \left(R_{xD} + R_{zD}\right)\right]^2 + \left(R_{yT} - R_{yD}\right)^2} \tag{3.53}$$

This singularity is associated with the angle set convention and is not dependent upon the particular robot geometry. Thus, the procedure for formulating the error function remains general with respect to the robot's geometry.

Using the rotational portion of the homogeneous transform ${}^0_n\mathbf{T}$ without extracting any particular angle set generates the trial solutions and associated errors without introducing a singularity. Simply formulate the error function using the nine rotational elements in ${}^0_n\mathbf{T}$ as shown in Equation (3.54).

$$E_O = \sqrt{\sum_{i=1}^{i=3}\sum_{j=1}^{j=3}\left(r_{ijT} - r_{ijD}\right)^2} \tag{3.54}$$

The scaling may also be simplified since the individual elements, r_{ij}, in ${}^0_n\mathbf{T}$ always vary between 1 and -1.

3.2.2 Exploration Patterns

An exploration strategy generates options during the direct search. Each option is called a trial solution. Many different exploration strategies can be formulated. They may be systematic, random, or both. There are also local and

global strategies. The direct search developed here is based on local explorations using the concept of a joint-level perturbation. The perturbations are performed in a pattern about the current base point of the search. Perturbation at the joint level simply means temporarily changing one or more of the joint displacements in either the positive or negative direction. The perturbations are only performed on the forward kinematic equations, and not on the actual robot. This chapter presents three different systematic exploration strategies. Chapter 4 discusses a strategy with a degree of randomness, called simulated annealing.

Simple exploration describes a strategy perturbing only one joint displacement at a time. This exploration pattern finds the individual influence of each joint on the end-effector motion at any given base point during the search. The simple pattern is fast computationally, but only explores a limited portion of the locality around the current base point.

Table 3.1 Three local exploration patterns and some of their attributes.

Exploration Pattern	**Attributes**
simple	fast finds individual influence of each joint explores only a limited portion of the locality
factorial	based on factorial design principles finds combinational effects somewhat slower than simple exploration much faster than exhaustive exploration
exhaustive	guaranteed to locate a local minimum explores every possible direction computationally intensive

Another perturbation pattern is based on 2^n factorial design. Factorial design is a mature and interesting research area in its own right. It includes far more than just 2^n exploration patterns. However, for the purposes of this discussion, the pattern based on 2^n factorial design is called factorial exploration. The factorial exploration pattern consists of all combinations of perturbed joint displacements. It does not generate combinations including unperturbed joint displacements. The factorial pattern explores more of the locality than the simple pattern, but less than the exhaustive pattern described below.

The most thorough exploration strategy is called the exhaustive pattern. The exhaustive exploration pattern generates all combinations of perturbed and unperturbed joint displacements. It explores in every possible direction from the current base point. Of the three exploration strategies just discussed, the exhaustive pattern is the most thorough, but also the most computationally demanding. Section 3.2.4 shows that the exhaustive exploration strategy is the only one that guarantees the finding of a local minimum.

These three exploration patterns correspond to points on the faces, edges, and vertices of a hypercube with the current base point at its center. Each point represents a trial solution. The dimension of the hypercube is equal to the robot's number of degrees of freedom. Figure 3.1 shows an example for a robot with three degrees of freedom. The points on the faces of the cube correspond to the simple pattern and the points at the vertices of the cube correspond to the factorial pattern. The exhaustive pattern includes all of the points.

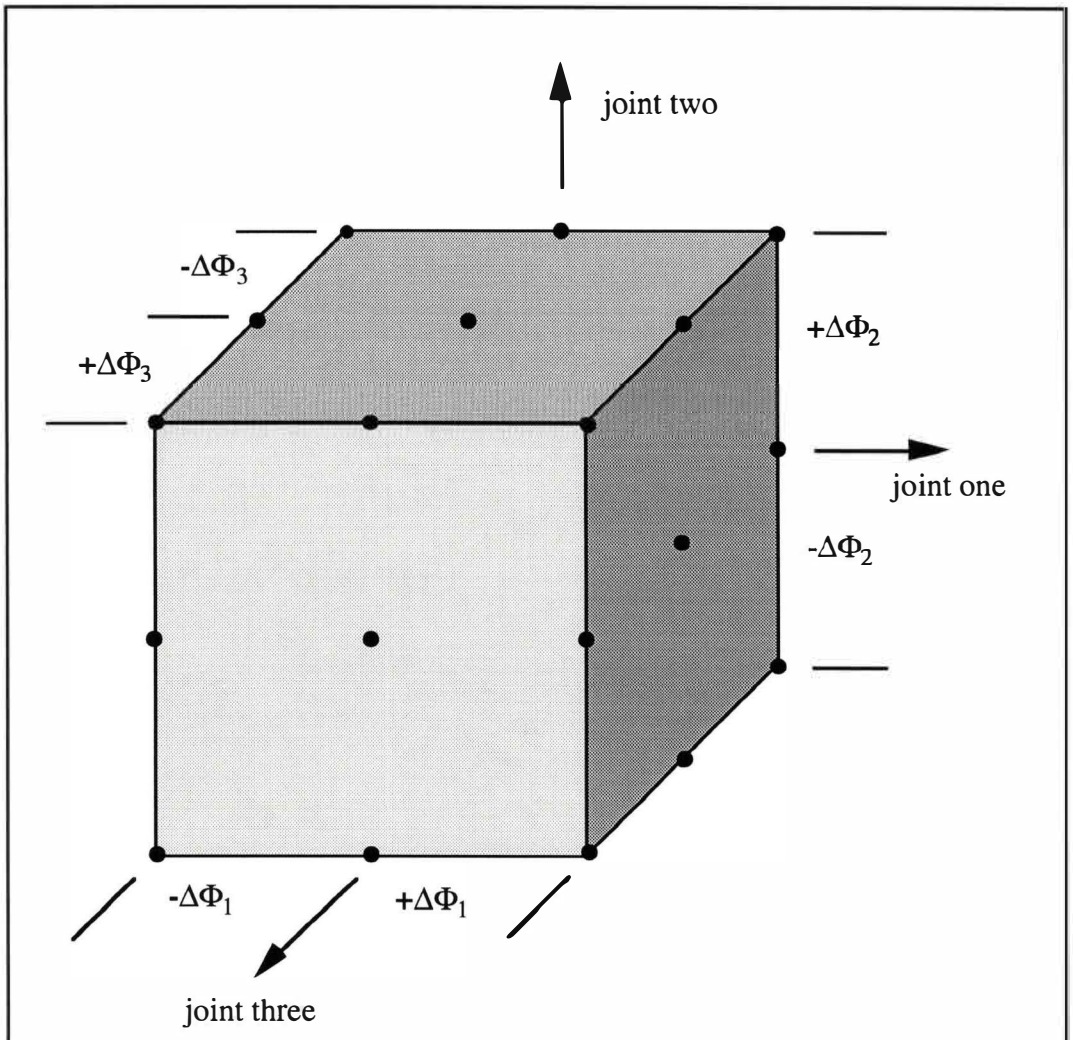

Figure 3.1 A three-dimensional hyper cube where the points on the faces represent the trial solutions for a simple exploration pattern, those on the vertices a factorial pattern, and all the points are generated in an exhaustive exploration.

3.2.3 Decision Making

A decision making strategy chooses one of the trial solutions as the next base point for the search. The decision is based on information calculated for

each of the trial solutions. A fully-constrained robot only requires calculating the Cartesian error. For a redundant robot, the decisions may be based on performance criteria as well. Three different decision making strategies are discussed below. These strategies may be applied to both redundant and fully-constrained robots. Chapter 4 discusses their application to redundant robots.

The steepest descent decision making strategy calculates the error for each of the trial solutions. This strategy is called steepest descent because it calculates the error for each of the trial solutions before choosing the minimum as the next base point.

An opportunistic decision making strategy often considerably increases the speed with which direct search finds the solution. In the opportunistic strategy, as soon as a trial solution decreasing the error is found, that solution is immediately chosen as the next base point and the search continues from there. This decreases the computational overhead by evaluating fewer trial solutions at each base point.

Table 3.2 Three decision making strategies and some of their attributes.

Strategy	**Attributes**
steepest descent	always chooses the minimum from a given set of trial solutions
	most demanding computationally
opportunistic	chooses a better solution as soon as one is found
	considerably increases speed of solution
pattern-move	incorporates the history of the search as part of the algorithm
	considerably increases the speed of solution
	effectiveness decreases as the non-linearity increases

The most famous of the direct search decision making strategies is the pattern-move due to Hooke and Jeeves (1961). The pattern-move strategy

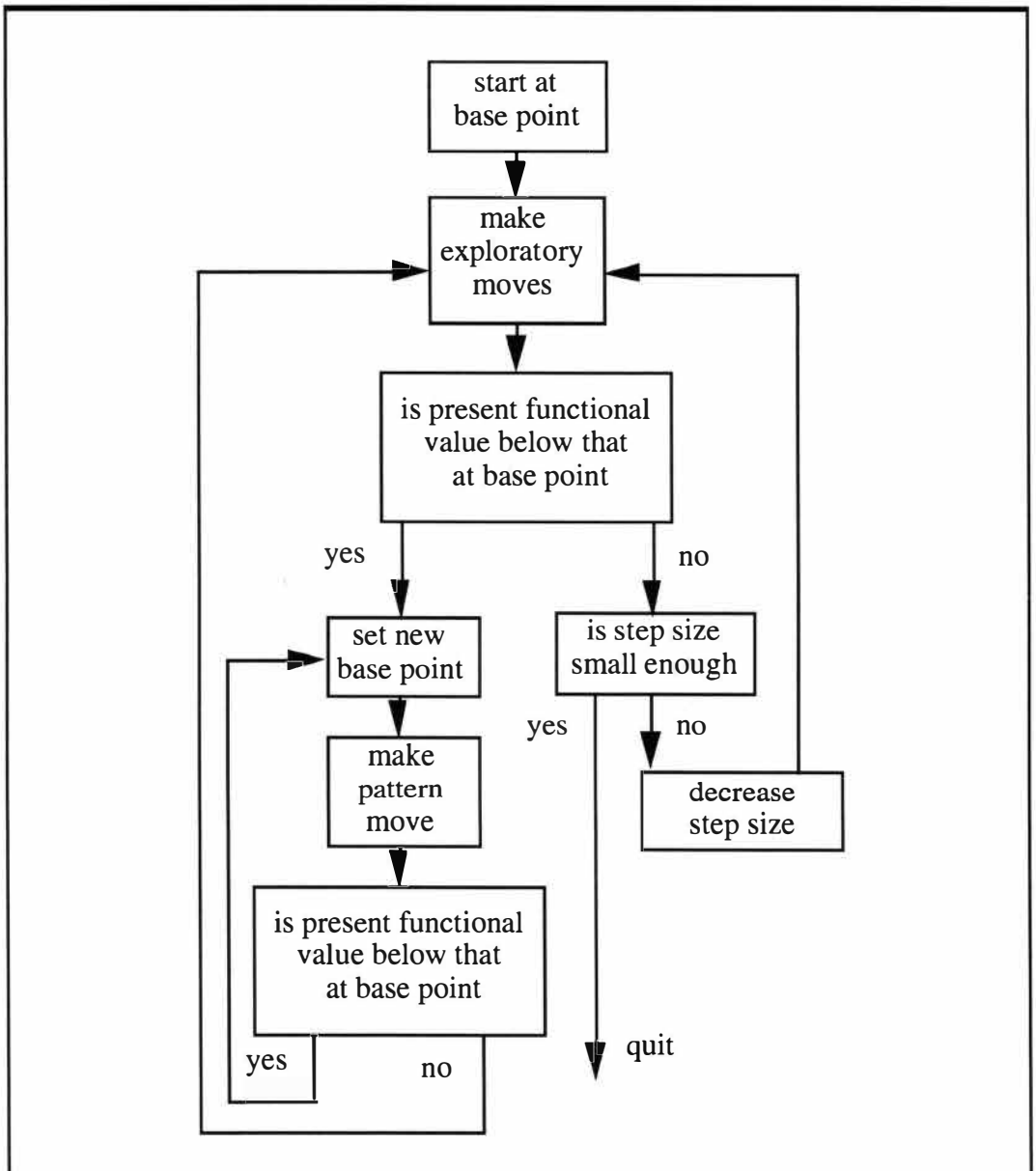

Figure 3.2 A flow chart for the pattern-move decision making strategy.

predicts which search direction likely leads towards a solution using a recent history of the search. Essentially, whenever a move is successful, that move is tried again. The movement towards the solution is the pattern-move. Figure 3.2 presents a flow chart of the algorithm, but the seminal manuscript of Hooke and Jeeves provides much more insight.

These three strategies make decisions based only on the first order (position level) information provided by the transformed end-effector constraints. The use of higher order geometric properties would certainly increase the convergence speed of the direct search. Thomas and Tesar (1982) formulated the first and second order geometric properties for a general serial robot in a concise fashion that explicitly shows the effect of each joint's motion on the resulting motion of the system. Using this information seems natural for an inverse kinematics strategy for general serial robots. The problem occurs at mathematical singularities. At these singularities the information provided by the higher order geometric properties becomes very difficult to interpret. Including the higher order properties in the basis for decision making would result in an algorithm that behaved differently at singularities. One of the goals for this inverse kinematics work was a method that would be equally effective throughout the robot's workspace.

Any search strategy must ultimately decide to terminate the search. Each of the direct search strategies discussed here repeatedly calculate an error between the trial and the actual solutions. Zero error (within an acceptable tolerance) is a reasonable condition for terminating the search. For an industrial robot, this could

require very small step sizes at the joint level. Beginning the search at a course resolution and then reducing the step size after bounding a solution is often advantageous. This suggests another condition for terminating the search that has physical relevance to robots under servo control at the joints. Simply terminate the search when the step size is smaller than the servo resolution.

Unfortunately, terminating the search without finding a solution is sometimes necessary. This may be because the search failed to find a solution, though one exists, or it may be because there is no solution. In either case, the search reaches an impasse from which it cannot find any better base point from which to continue. Chapter 4 develops several methods of adjusting scale factors to force the convergence of direct search algorithms. To prevent the algorithm from degenerating into an infinite loop, the scale factors may only be adjusted a finite number of times. After adjusting the scale factors several times, deciding to terminate the search without finding a solution may be the most reasonable choice.

3.2.4 Optimality

This chapter presents several direct search strategies. In each of them, a Cartesian error is formulated as an unconstrained objective function. Minimizing the function until its value is zero is the goal of the search. It is certainly reasonable to question which, if any, of the strategies guarantee minimization of an unconstrained objective function and exactly what sort of minimum is found. Remember that direct search is a discrete method which only finds minima to within the resolution of the search. Also remember that the direct search

strategies developed in this chapter employ local exploration patterns. Therefore, only local minimization will be discussed.

This section first defines the terms local minimum and isolated local minimum. The conditions such that a given set of joint displacements represents a minimum are then derived. Finally, it is shown that direct search with an exhaustive exploration pattern always finds a set of joint displacements which satisfies the conditions for a local minimum.

Definition 1: *A local minimum is a point such that all other points in the locality represent function values that are greater than or equal to the function value at the local minimum. Symbolically, this may be written as*

$$\Delta f(\Phi) = f(\Phi) - f(\overline{\Phi}) \geq 0 \tag{3.55}$$

where $\Delta f(\Phi)$ *represents the change in the function value,* $f(\overline{\Phi})$ *represents the function value at the local minimum and* $f(\Phi)$ *represents the function value at any other point in the locality.*

Definition 2: *An isolated local minimum is a point such that all other points in the locality represent function values that are strictly greater than the function value at the isolated local minimum. Symbolically, this may be written as*

$$\Delta f(\Phi) = f(\Phi) - f(\overline{\Phi}) > 0 \tag{3.56}$$

where $\Delta f(\Phi)$, $f(\overline{\Phi})$, and $f(\Phi)$ *are as defined above.*

The decision making strategies discussed in this chapter always choose a trial solution which decreases the value of the objective function if there is one. No provision was made for choosing trial solutions which represent equivalent function values and the search may terminate in their presence. Therefore, these

direct search strategies cannot possibly guarantee an isolated local minimum. The conditions under which the joint angle set $\overline{\Phi}$ meets the definition of a local minimum are now derived. The derivation begins with the familiar Taylor series expansion for a function of multiple variables.

$$f(\Phi) = f(\overline{\Phi}) + \nabla f(\overline{\Phi})^{T} \Delta\Phi + \frac{1}{2}\Delta\Phi^{T}\nabla^{2} f(\overline{\Phi})\Delta\Phi + \text{HOTS} \tag{3.57}$$

In this equation, $f(\Phi)$ represents the objective function, $\overline{\Phi}$ the expansion point, $\nabla f(\overline{\Phi})$ and $\nabla^2 f(\overline{\Phi})$ the first and second derivatives evaluated at the expansion point, and HOTS the higher order terms. Dropping the higher order terms and solving for changes in the objective function yields

$$\Delta f(\Phi) = \nabla f(\overline{\Phi})^{T} \Delta\Phi + \frac{1}{2}\Delta\Phi^{T}\nabla^{2} f(\overline{\Phi})\Delta\Phi \tag{3.58}$$

The first requisite for an extremum of a function is the stationary condition

$$\nabla f(\overline{\Phi}) = \mathbf{0} \tag{3.59}$$

Equation (3.59) is often called the necessary condition for a minimum. With the stationary condition, Equation (3.58) now becomes

$$\Delta f(\Phi) = \frac{1}{2}\Delta\Phi^{T}\nabla^{2} f(\overline{\Phi})\Delta\Phi \tag{3.60}$$

The sign of $\Delta f(\Phi)$ is very important and is determined by the right hand side of this equation. If it can be shown that $\Delta f(\Phi)$ is always greater than or equal to zero, then by Definition 1, the current expansion point is a local minimum. Notice that the right hand side of Equation (3.60) is in quadratic form. If the quadratic form is positive semi-definite, then by the definition of positive semi-definiteness, $\Delta f(\Phi)$ is greater than or equal to zero for all $\Delta\Phi$ and the current expansion point represents a local minimum.

The condition that $\nabla f(\overline{\Phi})$ must equal zero is treated first. Recall that the direct search inverse kinematics strategies test feasible directions by varying each of the joint displacements in Φ. The strategy is to move in the direction of decreasing function values, $f(\Phi)$, until there is no direction which further decreases $f(\Phi)$. Once the search is started, it continues in directions decreasing $f(\Phi)$ until finding no direction that further decreases $f(\Phi)$.

The term $\nabla f(\overline{\Phi})$ is the vector of first partial derivatives with respect to each of the joint displacements. Addressing this term requires searching the principle directions. The principle directions relate to changing each joint displacement individually while holding the others constant. Changing each joint displacement individually and calculating the function value generates a discrete analog of one element of the vector $\nabla f(\overline{\Phi})$. Both the simple and the exhaustive exploration patterns search the principle directions. That a search direction will not further decrease $f(\Phi)$ indicates that the function value experiences either a curvature reversal or a loss of curvature in that direction. Recall that curvature is defined at a point as the value of the derivative of the polar angle with respect to the arc length. Given that $f(\Phi)$ is continuous in the first derivatives, either a curvature reversal or a loss of curvature must be accompanied by the first derivative passing through zero.

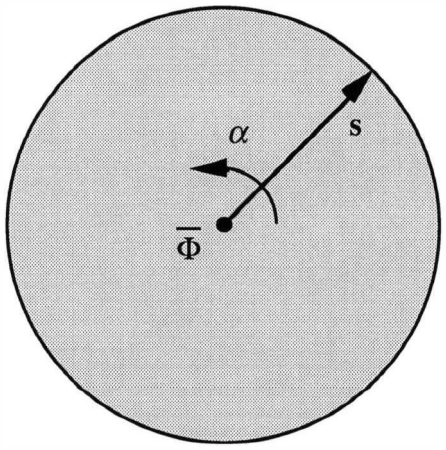

Figure 3.3 A stationary locality about $\overline{\Phi}$, in which α and s can be chosen so as to generate all points in the locality.

To show that direct search, at least with an exhaustive exploration pattern, finds a point where $\nabla^2 f(\overline{\Phi})$ is positive semi-definite consider the two dimensional case shown in Figure 3.3. Directions radiating from stationary point $\overline{\Phi}$ are generated by

$$\tilde{\Phi} = \overline{\Phi} + \alpha \mathbf{s}(\overline{\Phi}) \tag{3.61}$$

where α and $\mathbf{s}$ can be chosen so as to generate all points in the locality of $\overline{\Phi}$. Substituting Equation (3.61) into (3.60) gives

$$\Delta f(\Phi) = \frac{\alpha^2}{2} \mathbf{s}^{\mathrm{T}} \nabla^2 f(\overline{\Phi}) \mathbf{s} \tag{3.62}$$

which is in the quadratic form

$$Q(\mathbf{z}) = \mathbf{z}^{\mathrm{T}} \mathbf{A} \mathbf{z} \tag{3.63}$$

The quadratic form of Equation (3.62) is positive semi-definite if, for all Φ, the value of $f(\Phi)$ is greater than or equal to the value of $f(\overline{\Phi})$. Thus, to guarantee a

local minimum, all directions must be searched in order to generate all $f(\Phi)$ values in the locality. Only the exhaustive perturbation strategy searches all feasible directions. The direct search terminates when there are no feasible directions decreasing $f(\Phi)$, though there may be directions of equivalent values. This corresponds to the positive semi-definite case.

Thus it is shown that direct search with an exhaustive exploration strategy guarantees finding a local minimum of the objective function, $f(\Phi)$. An argument based on common sense is perhaps more convincing. If the search tests every feasible direction, and none decrease the function value, then the current point must be a local minimum. Without testing every feasible direction, direct search cannot guarantee that there isn't at least one direction that further decreases the function value.

Showing that direct search finds local minima also shows the condition under which direct search fails to find a solution to the inverse kinematics problem. Direct search may fail if there is a minimum directly between the current and the desired end-effector displacements. The physical interpretation of this situation is that the robot cannot move directly from where it is, to where it is commanded to go. This situation always occurs at workspace boundaries as the robot physically cannot move in the desired direction. The failure mode of the direct search is rather benign under these circumstances. The search terminates and is ready to find a new solution if given an attainable end-effector placement.

Unfortunately, there may be bounded local minima between the current and the desired end-effector displacements, even though both are attainable. This may result in the search terminating prior to finding a solution. Note that a

bounded local extremum is but one of the failure modes of the Newton-Raphson numerical solution method. A direct search typically deals with bounded local minima by changing scaling factors in order to change the shape of the objective function, $f(\Phi)$, and force convergence. Chapter 4 develops several methods for initially choosing scaling factors and for dynamically changing their values as the search progresses.

3.2.5 Relative Computational Complexity

The computational burden and the complexity of the computer code are reasonable considerations with any method of solving a problem on a computer. Unfortunately, it is difficult to assess the computational burden of most direct search strategies. This is because direct search often includes heuristics, such as the pattern move or the opportunistic decision making strategy, which may accelerate the solution speed. The effectiveness of these acceleration strategies influences the solution speed.

Without any heuristics, the relative computational burdens for the three perturbation methods (simple, factorial and exhaustive) are respectively: n, 2^n, and 3^n; where n represents the robot's number of degrees of freedom. However, even without the acceleration heuristics, it is possible that a more thorough exploration strategy will generate trial solutions leading more directly to a solution. For this reason, even without acceleration heuristics, the relative solution speed can only be estimated.

A direct search algorithm which solves the inverse kinematics for a general serial robot can be translated into computer code in a simple and concise

fashion. The program which generated the results presented in this dissertation is less than 300 lines of C computer code. This program includes the simple, factorial, and exhaustive exploration patterns as well as the steepest descent, opportunistic, and pattern-move decision making strategies. The code is general and requires only the appropriate Denavit and Hartenberg parameters to solve the inverse kinematics problem for any specific serial robot. The simplicity of the computer code for these direct search methods results from the repeated application of simple logic. The simplicity of the formulation suggests the possibility of implementing direct search inverse kinematics with specialized high-speed electronic hardware. In particular, hardware for performing the forward transformations that generate the error function would increase the speed of a direct search algorithm considerably.

3.2.6 Recommended Direct Search Applications

Since there are many different inverse kinematics methods, considering the conditions under which direct search techniques would be recommended is very important. These conditions may be discussed in terms of the generality of direct search inverse kinematics and its ability to converge at exactly singular positions of the robot.

This dissertation develops direct search techniques general in the sense that they will solve the inverse kinematics problem for a robot of any serial geometry. The exploration patterns and decision making strategies are formulated without geometric simplification or algebraic manipulation particular to any given

robot. This makes direct search inverse kinematics attractive when considering many different robot geometries, as is typically the case during robot design.

Table 3.3 Some positive and negative qualities of the direct search inverse kinematics method.

Positive Qualities	**Negative Qualities**
General with respect to geometry	slower than closed-form solutions
General with respect to objective function	slower than numerical methods which incorporate derivative information (except at, or near, singularities)
can be automatically formulated	
converges at singularities	
straightforward implementation	

A general approach to the inverse kinematics problem is essential to the application of a modular robot architecture. The generality of direct search inverse kinematics also allows its application with robots having complex geometries. This generality gives direct search inverse kinematics a distinct advantage over closed-form techniques in these application areas. However, other numerical methods of solving the inverse kinematics problem may also be formulated in a general fashion. For instance, an iterative method based on inverting the Jacobian is described in the literature as part of a reconfigurable and modular manipulator system (Kelmar and Khosla, 1990).

Direct search's ability to converge at exact singularities is unique among inverse kinematics methods, closed-form or numerical. Because all robots have singularities, this is an important recommendation for the continued development of direct search inverse kinematics. Direct search converges at singularities

because generating trial solutions and making decisions does not require any scalar or matrix inversion. The Newton-Raphson, predictor-corrector, or any method requiring derivatives, not only can't converge exactly to singularities, but also become ill-behaved near singularities. The difficulties associated with singularities is also the reason higher order geometric properties, such as those formulated by Thomas and Tesar (1982) for general serial robots, were not included as a basis for decision making in the strategies this chapter develops.