Chapter 2: Literature Review

Inverse kinematics is fundamental to robotics and has been extensively studied. Kinematicians whose mechanism theory provided elegant **closed-form*** solutions influenced the early years. Digital computers made the numerical solution of these complex nonlinear kinematic equations possible as well. During the 1970's, the search for an entirely general closed-form solution dominated inverse kinematics research. During the 1980's, the widespread availability of redundant research robots and powerful computers focused attention on optimization and **pseudoinverse** techniques. The robots of the 1990's will have more sensors, more redundancy, and be controlled by computers more powerful than ever before. The computers made by the Silicon Graphics company, for example, have shown a price-to-performance improvement of two hundered times in only five years. The inverse kinematics problem must now incorporate multiple performance criteria so that the robot can be in the best possible kinematic configuration while performing its task.

The synthesis of path-generating mechanisms is a close relative of the inverse kinematics problem. In both cases, kinematic displacements – link lengths for mechanism synthesis and joint displacements for inverse kinematics – are derived such that an end-effector will follow a prescribed path. By modeling a robot with six degrees of freedom as a mechanism with a mobility of one,

---

* The glossary defines words appearing in bold.

kinematicians applied mechanism analysis techniques developed during the 1950's to the inverse kinematics problem. Freudenstein and Roth are two researchers from mechanism analysis instrumental in the development of closed-form inverse kinematics. Though Freudenstein once declared the closed-form solution for a general robot with six degrees of freedom the "Mount Everest" of kinematic problems (Freudenstein, 1973), both Freudenstein and Roth published articles describing numerical solutions to the mechanism analysis problem; Freudenstein as early as 1959 (Freudenstein and Sandor, 1959)(Roth and Freudenstein, 1963). Other techniques from mechanism analysis have found application in robotics. **Kinematic influence coefficients** developed by Benedict and Tesar (1978) for the analysis of multi-degree of freedom planar mechanisms were later formulated for serial spatial robots by Thomas and Tesar (1982). The Kinematic influence coefficients are entirely general, applying to robots with any number of degrees of freedom. A physical plant description for the **dynamic model** is readily formulated in terms of Kinematic influence coefficients, which recommends their use for generalized inverse kinematics. A generalized inverse kinematics strategy may also exploit the sequential-filters method developed by Eschenbach and Tesar (1969) for mechanism synthesis.

A typical industrial robot has six degrees of freedom. This is because there are six spatial parameters necessary to specify the position and orientation of a rigid body in space. With the appropriate geometric relations the six independent variables, the joint displacements, are found in terms of the six parameters constraining the end-effector's displacement. Roth and other researchers have shown that there are a finite number of solutions to the inverse kinematics problem for a fully constrained robot (Roth et al., 1973). Much thoughtful research has been devoted to finding these solutions in closed form. This research generated books full of closed-form inverse kinematic solutions for a vast array of specific robots (Roth, 1979) (Duffy, 1980) (Angeles, 1982). Since closed-form solutions are computationally efficient, most industrial robots are designed to have known closed-form solutions. Modern applications of closed-form inverse kinematics include a robot designed for the Space Station Remote Manipulator System (Crane and Duffy, 1991).

Table 2.1 Several influential researchers, their interests, and some of their contributions to inverse kinematics.

Researcher	**Interest**	**Contributions to Inverse Kinematics**
Dimentberg 1940-1950	analysis of spatial mechanisms	dual number quaternions screw algebra framework for spatial mechanism analysis
Freudenstein 1950-1970	mechanism synthesis	loop equations vector analysis the "Mount Everest" declaration
Duffy 1970-present	6 DOF robots	general approach systematic and unified work closed solutions for most 6 DOF geometries
Whitney 1960-1980	human prostheses	resolved rate application of pseudoinverse

Numerical methods have long been applied to the inverse kinematics problem. The Newton-Raphson technique used by Uicker, Denavit and Hartenberg is generally cited as the first time numerical methods solved the inverse kinematics problem and is still in wide use. The predictor-corrector technique, also used for solving differential equations, can solve the inverse kinematics problem. Standard optimization software, such as Minpack 1, and standard numerical techniques, such as Powell's method, can also solve this problem.

Most researchers discuss singularities when reporting on inverse kinematics methods. Physically, a singularity occurs when the robot loses one or more inputs as independent controlling parameters. A common singularity arises when two revolute axis are collinear. There are also singularities at workspace boundaries. A robot with six degrees of freedom loses at least one degree of end-effector **dexterity** at singularities. All robots can exhibit singularities and it is, in general, impossible to avoid them by simply restricting the workspace (Maciejewski and Klein, 1988). Thus, a generalized inverse kinematics solution should be robust in the presence of singularities.

During the 1980's, attention began to shift towards kinematically redundant robots and the nature of the research changed dramatically. Redundant robots are typically able to achieve a given end-effector displacement with an infinite combination of joint displacements. This gives rise to the possibility of incorporating performance criteria in order to improve the robot's operation while satisfying the displacement constraints on the end-effector. Virtually all of the inverse kinematics methods for redundant robots are extensions of Whitney's "**resolved-rate**" work (Whitney, 1969). Rather than solving the position-level equations, the resolved rate method uses the **Jacobian** matrix while solving for the joint speeds. The Jacobian matrix relates the end-effector velocity to the joint speeds. The Jacobian matrix for redundant robots is not square because there are more joints than end-effector constraints. Integrating the joint speed gives the displacements. The Jacobian-based techniques typically fall into either the pseudoinverse or the extended Jacobian inverse category.

The pseudoinverse minimizes the two-norm of the vector of joint speeds. Though the pseudoinverse is very commonly used for redundant robots, there are some serious drawbacks. The simple-mindedness of only considering the joint speeds leads pseudoinverse solutions into a variety of traps. A common problem cited with the pseudoinverse is that it is not conservative. That is, when the end-effector follows a path that returns to its starting place, the joint displacements may not return to their starting place. This will cause the robot to drift in **joint space**, though still satisfying the end-effector constraints. The drift can often lead to a singularity. The drift is due to the loss of position-level information during the pseudoinverse of the Jacobian. It is entirely possible that the same two-norm of joint speeds can be achieved with the robot in an infinite number of different positions. This difficulty with position-level criteria has led to published suggestions for abandoning the pseudoinverse altogether if position-level criteria are to be incorporated in the solution (Carnigan, 1991) (Egeland et al., 1991). This is a very serious problem. Position-level geometry is fundamental to understanding the state of the robot. The very practical limits on joint displacements are at the position level as well. The pseudoinverse breaks down theoretically at singularities. Due to the finite numerical resolution of digital computers, the pseudoinverse also fails near singularities. Self-motion is the most common way researchers have attempted to alleviate the problems with the pseudoinverse. By choosing the self-motion, researchers have incorporated other performance criteria into the solution. Self-motion has been applied to singularity avoidance (Bedrossian and Flueckiger, 1991), kinetic energy minimization (Dubey and Lu, 1988), dexterity optimization (Martin and Satterlee, 1987), joint limit and obstacle avoidance (Nenchev, 1989), and the incorporation of task-independent criteria (Cleary and Tesar, 1990). Researchers have also used self-motion to alleviate the problems associated with an ill-conditioned Jacobian near a singularity (Maciejewski and Klein, 1988).

The extended Jacobian method is another solution to the inverse kinematics problem for redundant robots. Rather than using the pseudoinverse to a non-square Jacobian, the extended Jacobian method makes the Jacobian square by adding constraints. This method has been used for avoiding obstacles and joint limits (Baillieul, 1986). The extended Jacobian method also breaks down theoretically exactly at singularities and fails numerically near singularities. The additional constraints exacerbate the problem by adding algorithmic singularities to the robot's geometric singularities.

Criteria for performance evaluation are fundamental to the operation of advanced robots. Combined with sensory input, these criteria can be used for enhancing the performance of the robot. Cleary and Tesar (1990) have proposed a set of criteria that are based on the physical parameters of the robot and are independent of the task. These criteria do not need to be reformulated each time the robot performs a different task. Sensory information and performance criteria must be ranked, **scaled**, **normalized**, and fused for intelligent use by a generalized inverse kinematics algorithm.

2.1 MECHANISMS

For hundreds of years, mechanisms have transmitted cyclic forces and motions. Mechanisms are found in a variety of mechanical systems, including: sewing machines, typewriters, automobiles, computer drives, and advanced semiconductor process tools. The complex geometry of mechanisms and their practical applications have long fascinated engineers and mathematicians. Steady research over the course of several decades extended the theory of mechanisms from the planar case to the spherical case, until finally advancing to the spatial case. Ultimately, the theory of spatial mechanisms provided the analytical foundation upon which inverse kinematics for serial robots was built.

Mechanism theory is applied to inverse kinematics by modeling the robot as a spatial mechanism. Imagine a robot turning the doorknob of a closed door. The doorknob represents a revolute joint. The base of the robot is connected to the ground, and so is the door. As it is turning the doorknob, the serial robot actually becomes a closed-loop spatial mechanism. A robot with six degrees of freedom has just enough dexterity to put a pure rotation on the doorknob. Thus, a robot with six degrees of freedom corresponds to a spatial mechanism with mobility one.

2.1.1 Dimentberg's Contributions

Dimentberg's application of dual-number quaternions and screw algebra to the analysis of spatial mechanisms during the late 1940's is seminal in the field of inverse kinematics. This work laid the mathematical foundation for much of the mechanism analysis that was later applied to the inverse kinematics of serial robots. The interested reader is directed to the original works of Dimentberg and others (Dimentberg, 1948) (Yang and Freudenstein, 1964).

A quaternion is a set of four real numbers: d, a, b, and c. These numbers are ordered and associated with four units: +1, i, j, and k. The unit +1 has the same properties as the real number 1. The properties of i, j, and k are:

$$i^2 = j^2 = k^2 = -1 \tag{2.1}$$

and

$$ij = k, \; ji = -k \tag{2.2}$$

The quaternion q is written:

$$q = d + ai + bj + ck \tag{2.3}$$

The i, j, and k units are essentially the base vectors of a Cartesian frame.

A screw, **L**, is simply an axis in space associated with a pitch magnitude. The screw coordinates are a set of six numbers defining the axis and the pitch.

$$\mathbf{L} = \left(L_1, L_2, L_3, L_4, L_5, L_6\right) \tag{2.4}$$

A screw system is closed under addition and multiplication. That is, the sum or product of any sets of screws in the system will remain in the system.

Dual-number quaternions and screw algebra provided a framework for analyzing mechanisms. This framework facilitated the extension of mechanism analysis from the plane to the sphere and ultimately from the spherical to the spatial case.

2.1.2 Sequential Filters

Eschenbach and Tesar (1969) proposed sequential filters for reducing the design space of the coplanar mechanism synthesis problem. More recently, Cleary and Tesar (1990) incorporated multiple performance criteria in a solution of the inverse kinematics problem for redundant robots using sequential filters.

Eschenbach and Tesar's application of sequential filters to mechanism synthesis specified an initial criterion, all linkages satisfying the motion of a coupler point through four point locations, to generate as many as 60,000 different linkages. This large design space was sorted by means of zones and given a design criterion number to reduce the solution space and provide the designer with a manageable range of solutions.

The concept of sequential filters is general and may be applied to the inverse kinematics problem for redundant robots. Eschenbach and Tesar's manuscript classified the design zones as necessary or desirable. In the context of the inverse kinematics problem, meeting the end-effector constraints is necessary, while optimizing performance criteria is desirable. The necessary conditions reduce the range of solutions. The final solution is chosen according to the desirability. Chapter 4 of this dissertation includes a more thorough discussion of the application of sequential filters to the generalized inverse kinematics problem.

2.1.3 Kinematic Influence Coefficients

A generalized inverse kinematics implementation must include criteria based on joint torques and forces. Calculating these torques and forces requires dynamic modeling and analysis. Kinematic influence coefficients are a concise vehicle for formulating both the dynamic model and the necessary computational sequence. The controlling equations explicitly maintain the system parameter and input/output dynamics.

Benedict and Tesar (1978) developed kinematic influence coefficients as a generalized means of modeling an N degree of freedom planar mechanism. Their model formulation quantitized the mechanism geometry in terms of Assur subgroups. As formulated, their model includes the analysis of complex mechanisms with multiple inputs. The kinematic influence coefficients are the heart of the formulations. These coefficients describe the total influence the N inputs have on the motion of the mechanism and allow a direct statement of the complex and coupled nonlinear differential equations controlling the response of the system. The complete system model includes externally applied loads, system mass, compliance and damping terms. Later, a similar modeling and computational technique was applied to serial robots.

Thomas and Tesar (1982) formulated a dynamic rigid-link model for serial robots using kinematic influence coefficients. The model formulation is entirely general and based solely on the geometry and physical properties of the robot. The model expresses the robot dynamics referenced to the joints of the robot.

2.1.4 Freudenstein's Contributions

Ferdinand Freudenstein greatly influenced the course of inverse kinematics research. A brief accounting of his work on the displacement analysis of mechanisms during the 1950's and 1960's, and its transition to inverse kinematics during the early 1970's, provides a unique insight into the foundations of the inverse kinematics.

During the 1950's Freudenstein was working on the synthesis of path-generating mechanisms for motion conversion and force transmission. Several of the ideas he published continue to permeate inverse kinematics research. In particular, the insistence on finding all existing solutions to a given problem is still pervasive; so much so that one respected introductory robotics text will not even admit that an algorithm has solved the inverse kinematics problem unless it is guaranteed to find all of the possible solutions (Craig, 1989). During the latter part of the fifties, Freudenstein worked on the synthesis of path-generating mechanisms using vector analysis. The equations were formulated so as to be solved by computer (Freudenstein and Sandor, 1959). Although a computer was used, the equations were still solved in closed form without **iteration**. During this time, Freudenstein espoused the idea of developing kinematic tools for use by a designer of mechanisms. The idea of using a set of tools for inverse kinematics analysis was also common during the 1970's and 1980's.

The geared five-bar was a mechanism that attracted Freudenstein's interest during the 1960's. Freudenstein's attempts at analyzing this mechanism produced equations with “a degree of complexity which renders closed-form algebraic solutions unobtainable” (Freudenstein and Roth, 1963). This forced Freudenstein to resort to an iterative Newton-Raphson numerical solution procedure. The Newton-Raphson procedure remains to this day a common method of solving the inverse kinematics problem. In 1964 Yang and Freudenstein applied dual-number quaternion algebra to the analysis of spatial mechanisms. The dual-number quaternion formulation allowed the derivation of explicit closed-form algebraic expressions solvable without iteration. At this time, Freudenstein expressed his opinion that, though complex, dual-number quaternion algebra could eventually become a tool for the displacement analysis of spatial mechanisms.

In the early 1970's, Yuan, Freudenstein, and Woo applied screw coordinates to the analysis of spatial mechanisms. They believed that screw algebra could solve for the displacements, velocities, accelerations, and forces of a general 7 link spatial mechanism (Yuan et al., 1971). The 7 link spatial mechanism represents the transition where mechanism analysis also includes serial robots. For the general seven-link spatial case, Freudenstein was forced to resort to numerical integration to solve for the displacements.

It was in 1972 that Freudenstein delivered an address entitled "Kinematics: Past, Present and Future" at the 12th Mechanisms Conference of the American Society of Mechanical Engineers. This address (and its subsequent publication) influences inverse kinematics research to this day. In it, Freudenstein declared the "closed form, algebraic solution of the input/output displacement equation of the general, single-loop, single-degree-of-freedom spatial mechanism consisting of seven links and seven turning pairs of arbitrary orientation in space" the "Mount Everest" of kinematics problems. Since then, researchers (to be discussed) have devoted their careers to finding this solution in closed form. Freudenstein also called for the development of a modern kinematics textbook similar in spirit to the seminal "Strength of Materials" by Timoshenko. In the following decade, at least four such textbooks appeared (Roth, 1979) (Duffy, 1980) (Angeles, 1982) (Barton, 1984).

In closing, as Freudenstein concentrated on mechanisms of increasing complexity: from four-bar, to five-bar, to the spatial 7-bar, so too did the complexity of his mathematical tools increase. Ultimately these mathematical tools became too complex for the design engineer. In Freudenstein's address at the mechanisms conference he noted that "computational capabilities of an order of magnitude and speed not heretofore attainable" have made complex design calculations possible. The computational capabilities available to the design engineer have increased at least 10,000-fold since then.

2.2 Fully-constrained Robots

Inverse kinematics methods for fully-constrained robots may be loosely classified as either closed-form or numerical. The closed-form methods are computationally very fast and execute in a repeatable and fixed amount of time. These qualities facilitate the application of closed-form methods in real-time control schemes. Numerical methods are slower, but they are much more general. Generality is recommended when a large number of options must be considered, such as is the case during robot design or in the reconfiguration process following a fault in one of the robot's components.

Table 2.2 Some positive and negative qualities of several inverse kinematics methods for fully-constrained robots.

Method	**Positive Qualities**	**Negative Qualities**
closed-form	finds all solutions very fast consistent speed	must be reformulated for each geometry singularities
Newton-Raphson	relatively general with respect to geometry	singularities non-convergent cycles
predictor-corrector	relatively general with respect to geometry fast for a numerical method	singularities difficulties with highly non linear equations

table look-up	very fast reliable	must be retabulated for each geometry vast amounts of memory requirements

2.2.1 Duffy's Contributions

When Freudenstein declared the closed-form solution of the inverse kinematics problem the "Mount Everest" of kinematics problems, Joseph Duffy was listening. This was a beautifully defined non-linear problem that Duffy intended to solve. In that same address Freudenstein called for a modern kinematics textbook "similar in spirit to the great works in the strength of materials." Duffy also took these words to heart and in 1980 published a book detailing his systematic development of "a unified theory for the kinematic analysis of mechanisms and manipulators (robotic arms)" (Duffy, 1980).

By all accounts, Duffy's work is exemplary. It is systematic and always intends to be general. Duffy's methods find all existing solutions to a given problem. Algebraic solutions, as opposed to iterative solutions, provide insight into various aspects of linkages, such as: overclosure, rotatability, type, and transmission characteristics. Finally, analytical solutions, when they are available, are typically much faster to solve on a computer than their iterative counterparts.

Unfortunately, the inverse kinematics problem turned out to be more difficult than Duffy imagined (Duffy, 1980). Even the nomenclature is demanding and somewhat confusing. Most researchers in the field of robotics hoped that Duffy would ultimately provide analysis tools for solving the general inverse kinematics problem. Unfortunately, the tools became so complex that it takes a kinematics expert just to use them. The algebraic equations are so complicated that even basic aspects, such as singularities, are difficult to discern. And finally, modern robots are beginning to exhibit serial kinematic redundancy. The inverse kinematics problem has changed considerably. It is no longer desirable to determine all existing solutions, but rather, the problem is to intelligently choose one. This is certainly not to say that Duffy's work is no longer relevant. Duffy and his students continue to develop their inverse kinematics methods and show that these methods apply to modern robots.

Crane, Duffy and Carnahan (1991) published a kinematic analysis of the space station remote manipulator system including inverse kinematics for the arm. This work serves as an excellent example of the applicability of Duffy's methods to modern redundant robots. Their approach chooses a desirable angle for one of the joints (either the 1st, 2nd or 7th), locks that joint, and then solves for the six remaining joint angles using Duffy's method. While there is some merit to this approach, especially given the limited computational capacity allotted for any one activity on the space station, it also illustrates how resolute Duffy and his associates are about solving the inverse kinematics problem in closed-form. Their approach could have easily been extended to allow numerical iteration on one of the joints, instead of locking, so as to optimize a performance criterion while still solving the remainder of the inverse kinematics problem in closed-form. This is probably an untenable solution to them. Fortunately, this type of approach is available to other researchers and will likely be exploited in the future.

Duffy's work is inarguably of the highest quality. The analytic closed-form solution to the general serial robot with six degrees of freedom continues to be of theoretical interest. It is also of practical importance to the application of both fully-constrained industrial robots and redundant robots.

2.2.2 Newton-Raphson Numerical Method

The Newton-Raphson technique as applied by Uicker, Denavit and Hartenberg (1964) is generally cited as the first time numerical methods were used for solving the inverse kinematics problem for a robot with six degrees of freedom. This method is still in wide use and papers describing minor modifications to the algorithm continue to be published (Gupta and Kazerounian, 1985). The popularity of the algorithm is due to its relatively simple formulation and quadratic rate of convergence. A formulation of the Newton-Raphson method and a description of several of its failure modes follows.

The Newton-Raphson method is essentially a root-finding scheme. It is applied by rewriting the inverse kinematics problem as

$$f(\Phi) - \mathbf{x} = \mathbf{0} \tag{2.5}$$

Because it is easiest to visualize the method in one dimension, the remainder of this discussion will focus on the one-dimensional case. The extension to multiple dimensions is straight-forward. Geometrically the Newton-Raphson method extends the tangent at an initial estimate of the solution, x_i, until the extension crosses zero. The point of the zero crossing is chosen as the next estimate of the solution, x_{i+1}, and the tangent at this point is again extended until it crosses zero. This process is repeated until the zero point is found to the desired level of accuracy. The Newton-Raphson algorithm is formulated via the Taylor series expansion of the function about the current iterative point

$$f(\Phi+\Delta\Phi)=f(\Phi)+f'(\Phi)\Delta\Phi+f''(\Phi)\frac{\Delta\Phi^2}{2}+\text{HOTS} \tag{2.6}$$

where HOTS represents higher order terms. Rewriting Equation (2.6) omitting the terms of order two and higher

$$f(\Phi+\Delta\Phi)=f(\Phi)+f'(\Phi)\Delta\Phi \tag{2.7}$$

Since a zero-crossing is being sought

$$f(\Phi+\Delta\Phi)=0 \tag{2.8}$$

and

$$\Delta\Phi=-\frac{f(\Phi)}{f'(\Phi)} \tag{2.9}$$

The most obvious failure of this method occurs when $f'(\Phi)=0$, which is precisely the case at a singularity of the robot. Because of limited accuracy in the computer representation of numbers, the method fails near mathematical singularities as well. Other failure modes of the Newton-Raphson method are less obvious but well-known and are found in most books on numerical methods.

One important failure mode occurs when a local extremum is between the current solution and the actual solution. Another failure mode occurs when the method degenerates into a non-convergent cycle.

The Newton-Raphson method enjoys wide application to the inverse kinematics problem in spite of these failure modes. As will be shown, the only failure mode of the direct-search method proposed in this dissertation occurs when a local extremum is encountered before an actual solution is reached. This is but one of the failure modes of the Newton-Raphson method. The direct-search method does not share either of the other failure modes: instability at singularities or non-convergent cycling.

2.2.3 Predictor-Corrector Numerical Method

The predictor-corrector method is another common numerical scheme for solving the inverse kinematics problem for robots with six degrees of freedom. The predictor-corrector method appears in numerous textbooks dating back at least thirty years. As might be inferred from the name, the predictor-corrector method has two parts: the predictor and the corrector. The predictor combines a polynomial curve fitting routine with several equally-spaced past values of the function to extrapolate, or predict, the next point. Information, including derivative information, at this new point corrects the prediction until it is within the desired tolerance. A smooth function will result in a good prediction and good overall performance. Unfortunately, if the function is rapidly changing, the prediction will likely be far removed from the actual value and the predictor-corrector method could diverge from the actual solution. There are also problems with starting and stopping the method. Because the corrector portion of the method uses information about the function's derivatives, the predictor-corrector method is unable to converge at singularities.

Since the predictor portion of the predictor-corrector method requires several previous values, another method of solving the inverse kinematics problem must start the algorithm. If another reliable method of solving the inverse kinematics problem must be available for starting the predictor-corrector algorithm, why bother using the predictor-corrector method at all? The response to this question is that the predictor-corrector method is 5-15 times faster than the Newton-Raphson method once it is started (Gupta and Kazerounian, 1985). Whether this increase in performance is worth the added complexity of dealing with two distinct inverse kinematics methods is certainly debatable. There is also a problem with stopping the predictor-corrector algorithm. Since the predictor relies on equally spaced previous steps, a step will not necessarily fall directly upon the desired termination point. Some other method that does not rely on equally spaced step sizes must terminate the predictor-corrector algorithm.

As with any numerical scheme that relies on derivative information, the predictor-corrector method theoretically breaks down at singularities. The predictor-corrector method also produces inaccurate results near singularities. Gupta and Kazerounian (1985) report errors at the joint-level from five to twenty-five degrees.

2.2.4 Modified Newton-Raphson and Modified Predictor-Corrector

Both the Newton-Raphson and predictor-corrector methods theoretically break down at singular configurations of the robot. Because of practical considerations, both of these methods also fail near mathematical singularities. Because of this, researchers applying Newton-Raphson and/or predictor-corrector methods to the inverse kinematics problem developed modifications to the original methods that improve their performance near singularities. The modifications, however, introduce an additional failure mode.

The addition of a strict-descent criterion improves the performance of the Newton-Raphson and predictor-corrector methods near singularities. This criterion is composed of two error measures, one translational and rotational. The modified algorithms only allow intermediate solutions which decrease the errors.

Although the modified Newton-Raphson and predictor-corrector algorithms perform significantly better than their unmodified counterparts near singularities, the strict descent criterion introduces another failure mode. This mode occurs if the intermediate solutions encounter a position-level extremum. This failure mode is not typically mentioned, although it will almost certainly occur if these algorithms are applied to an actual robot. These modifications only improve the performance near singularities, and neither the modified Newton-Raphson nor the modified predictor-corrector will converge at exact singularities.

2.2.5 Table Look-up

Table look-up is another method of solving the inverse kinematics problem (Everett and McCarroll, 1986). This method is appealing in the sense that it is very fast and has no problems with singularities. The concept is quite simple, though it requires either another inverse kinematics solution method or some way of measuring the end-effector position of the actual robot. Simply cycle the robot, or the simulated robot, throughout its workspace and record the set of joint angles corresponding to each position of the end-effector.

Though undeniably fast, the table look-up method has a major problem with memory requirements. Given a six-revolute robot with joint travels of 180 degrees, the memory requirement for a resolution of 1 degree is 180^6 (3.4×10^{13})

bytes. This is at least four orders of magnitude greater that the storage capacity of current magnetic media, though not inconceivable for optical media. Most industrial robots have a joint level resolution of at least .001 degrees which increases the storage requirements to at least 6.8 X 10^{31} bytes. This is far beyond the capacity of any current memory storage device. Combining an interpolation scheme with a lower resolution table reduces the memory requirements but cannot guarantee accuracy.

2.3 REDUNDANT ROBOTS

The introduction of robots with serial kinematic redundancy fundamentally changed the nature of inverse kinematics research. For redundant robots, the inverse kinematics problem becomes very much an optimization problem. The desired placement of the end-effector represents equality constraints for the optimization.

2.3.1 Whitney's Contributions

Daniel Whitney introduced resolved motion rate control in 1969. His approach serves as the basis for virtually all the inverse kinematics methods developed for redundant robots since that time. Whitney was working on human prosthetic arms for forequarter (no arm stump at all) amputees. There was no arm left from which to measure a physical displacement as a controlling signal for the prosthesis. Only EMG signals (low-level electrical signals generated by active muscles) were available which Whitney suggested could specify a rate of change for the prosthetic arm's displacement. This represents the "rate control" portion of Whitney's "resolved motion rate control". The "resolved motion" comes from his speculation that it would be easier for the amputee to control the prosthesis in world, rather than arm, coordinates. Hence, in Whitney's description, "the motions of the various motors are combined and resolved into separately controllable hand motions along world coordinates" (Whitney, 1969). This is a statement of the inverse kinematics problem at the velocity level.

Whitney's formulation of the problem is extremely simple. The world coordinates, represented by the vector $\mathbf{x}$, are related to the joint displacements, represented by the vector Φ, by some function

$$\mathbf{x} = f(\Phi) \tag{2.10}$$

As Whitney was interested in the rate of change of $\mathbf{x}$, he differentiated this equation to obtain

$$\dot{\mathbf{x}} = \mathrm{J}(\Phi)\dot{\Phi} \tag{2.11}$$

where $\mathrm{J}(\Phi)$ represents the Jacobian (the matrix of first derivatives of $f(\Phi)$). After that, he simply solves for $\dot{\Phi}$ given $\dot{\mathbf{x}}$

$$\dot{\Phi} = \mathrm{J}^{-1}(\Phi)\dot{\mathbf{x}} \tag{2.12}$$

This simple three line formulation served as the basis of virtually all redundant inverse kinematics for twenty-five years. Whitney even formulated the problem for the redundant case via a pseudoinverse.

While Whitney's formulation is concise and seductive in its simplicity, it is not without major limitations; the most obvious being when the Jacobian is rank-deficient. Another glaring deficiency is the neglect of position-level information by only considering the Jacobian. Position-level considerations are indispensable for actual implementations, which must consider joint travel constraints, obstacles in the environment, and **conservative motion** to name but a few.

These criticisms are not intended to diminish Whitney's contributions. The simple resolved motion rate control was perfectly reasonable given the computational power available during the 1960's. Also, since Whitney was working with forequarter amputees, he had a physical motivation for using rates as the controlling inputs. These criticisms should, however, be noted by the hundreds of researchers that have jumped on the resolved-rate bandwagon during the 1980's and 1990's. There is a richness and wealth of information at the position-level that should not be ignored.

2.3.2 Pseudoinverse

The pseudoinverse is by far the most common method of solving the inverse kinematics problem for redundant robots. Hundreds of papers detailing this method have been published, beginning when Whitney discussed the pseudoinverse in his original resolved rate paper. This section includes a definition of the pseudoinverse, a method for calculating the pseudoinverse, and a method for applying it to the inverse kinematics problem. The myriad of problems which plague pseudoinverse techniques are also discussed.

A n x m pseudoinverse, $\mathbf{A}^+$, of an n x m matrix, $\mathbf{A}$, satisfies the four Penrose conditions:

$$\begin{aligned} \mathbf{A}\mathbf{A}^+\mathbf{A} &= \mathbf{A} \\ \mathbf{A}^+\mathbf{A}\mathbf{A}^+ &= \mathbf{A}^+ \\ \left(\mathbf{A}\mathbf{A}^+\right)^T &= \mathbf{A}\mathbf{A}^+ \\ \left(\mathbf{A}^+\mathbf{A}\right)^T &= \mathbf{A}^+\mathbf{A} \end{aligned} \qquad (2.13)$$

If n > m, as is the case with redundant robots, then the pseudoinverse is calculated simply as

$$\mathbf{A}^{+} = \mathbf{A}^{T}\left(\mathbf{A}\mathbf{A}^{T}\right)^{-1} \tag{2.14}$$

The application of the pseudoinverse to the inverse kinematics problem is via Whitney's resolved rate technique. Beginning with the resolved rate formulation

$$\dot{\mathbf{x}} = \mathrm{J}(\Phi)\dot{\Phi} \tag{2.11}$$

where $\mathrm{J}(\Phi)$ represents the Jacobian of $f(\Phi)$. Dividing both sides of Equation (2.11) by $\mathrm{J}(\Phi)$ gives the joint speeds.

$$\dot{\Phi} = \mathrm{J}^{-1}(\Phi)\dot{\mathbf{x}} \tag{2.12}$$

Since the Jacobian for a redundant robot is not square, the pseudoinverse, $\mathrm{J}^{+}(\Phi)$, is used instead of $\mathrm{J}^{-1}(\Phi)$

$$\dot{\Phi} = \mathrm{J}^{+}(\Phi)\dot{\mathbf{x}} \tag{2.15}$$

The simplicity of this formulation indicates that there might be some problems in applying it to something as complex as a redundant robot. The pseudoinverse as shown minimizes the two-norm of the joint speeds. This will be derived in the Chapter 3. Unfortunately, the minimum two-norm of joint velocities does not necessarily correspond to improved or even acceptable performance. If minimizing the two-norm of the joint velocities is the only criterion, better performance would likely be attained with a fully-constrained robot.

Since the pseudoinverse minimizes the two-norm of joint velocities, researchers originally thought that it would naturally avoid singularities. This is because singularities are often accompanied by very high demands on the joint velocities. As has since been realized, the pseudoinverse often leads the robot into singular configurations (Carnigan, 1991). This is a problem since the pseudoinverse fails at singularities.

One of the many problems with using the pseudoinverse for an actual robot is that physical constraints are not addressed. These constraints are at the position, velocity and acceleration levels. Anyone using an actual robot experiences the position-level constraint on joint travels (joint limits). The pseudoinverse does not address this very practical constraint. The top speed of the actuators is a constraint at the velocity-level. The pseudoinverse comes closest to addressing this constraint, though many researchers have noted that the pseudoinverse often makes unrealistic demands on the speed capabilities of the actuators. Finally, acceleration constraints relate to the finite ability of the actuators to produce forces or torques. The pseudoinverse does not address acceleration constraints.

The most common problem cited with the pseudoinverse is that it does not produce conservative motion. That is, following a closed path in **end-effector space** will not necessarily result in a closed path in joint space. In effect, the robot drifts in joint space. Numerous researchers studying this problem found that it is unpredictable and often leads to instabilities. It is easy to understand the origins of the non-conservative drift. The drift occurs at the position level, while the pseudoinverse is applied at the velocity level. It is altogether possible that the same two-norm of joint speeds produce different position-level results. A non-conservative path is unpredictable at the joint-level. This adversely affects repeatability in manufacturing operations.

A final problem with the pseudoinverse relates to unit invariance. That is; the pseudoinverse will give different results depending upon the units representing the dimensions of the robot and the task. For instance, measuring lengths in feet instead of inches changes the results of the pseudoinverse. The physical meaning of the problem does not change, yet changes in parameterization yield different solutions (Lipkin, 1990). This is a fundamental consideration in inverse kinematics. It occurs in the very formulation of the problem: both translational and rotational coordinates describe the placement of the robot's end-effector. There are essentially three ways of handling this problem. The first is to ignore it and simple-mindedly mix units. This is what happens in the pseudoinverse. Careful algebraic manipulation may maintain separation between rotational and translational terms (Duffy, 1990). This method, however, severely limits the number and type of performance criteria that may be considered in the solution. Finally, systematic scaling procedures will provide a rational basis for comparing rotational and translational properties (Bevill and Tesar, 1990).

Considering the myriad of problems associated with the pseudoinverse as a practical solution to the inverse kinematics problem, it is difficult to understand why research on this method continues.

2.3.3 Extended Jacobian

The extended Jacobian method for solving the inverse kinematics problem for redundant robots receives considerable attention in the literature. The method is also applied to actual robot arms at the Jet Propulsion Lab in Pasadena and is the current choice for the redundant Ranger arm at the University of Maryland. Conceptually the extended Jacobian method is quite simple. For a redundant robot with n joints and m end-effector constraints, n is greater than m and the

Jacobian has more columns than rows. The extended Jacobian method makes the Jacobian square by adding an additional n minus m constraints. Inverting the extended Jacobian gives the joint speeds. Unfortunately, the addition of constraints adds algorithmic singularities that may prohibit inversion.

Mathematically the extended Jacobian method is not difficult to formulate. Simply define n - m constraints

$$G_1(\Phi)=0,\, G_2(\Phi)=0,\ldots,G_{\text{n-m}}(\Phi)=0 \tag{2.16}$$

and add them to the forward kinematic relations of the robot

$$\begin{bmatrix} f(\Phi) \\ G_1(\Phi) \\ \vdots \\ G_{\text{n-m}}(\Phi) \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.17}$$

From here the method proceeds in the usual resolved rate fashion. Differentiating both sides of Equation (2.17) gives

$$\begin{bmatrix} \dfrac{\partial f}{\partial \Phi} \\ \vdots \\ \dfrac{\partial G}{\partial \Phi} \end{bmatrix} \dot{\Phi} = \begin{bmatrix} \dot{\mathbf{x}} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{2.18}$$

The square matrix, $\mathbf{J_e}$,

$$\mathbf{J}_{\text{e}} = \begin{bmatrix} \dfrac{\partial f}{\partial \Phi} \\ \vdots \\ \dfrac{\partial G}{\partial \Phi} \end{bmatrix} \tag{2.19}$$

is called the extended Jacobian.

Dividing both sides of Equation (2.18) by the extended Jacobian gives the joint speeds.

$$\dot{\Phi} = \mathbf{J}_e^{-1}\begin{bmatrix}\dot{\mathbf{x}}\\ \mathbf{0}\end{bmatrix} \tag{2.20}$$

The extended Jacobian method is mathematically simple and conceptually appealing in the sense that constraints may be chosen to optimize various performance criteria. There is, however, a serious singularity problem with the method. This is because adding constraints adds algorithmic singularities to the geometric singularities of the robot. Typically, researchers use the self-motion of the robot and a singularity detection scheme to avoid the singularities. This begs one to question the advantage of using a redundant robot that is devoting its redundancy to avoiding singularities that were artificially introduced by the inverse kinematics algorithm. Unfortunately, the more performance criteria that are added, the more complex the singularity problem becomes. This essentially eliminates the extended Jacobian method as a solution to the generalized inverse kinematics problem.

Since the extended Jacobian method is a velocity-level scheme, it also exhibits non-conservative motion and the associated joint-level drift. In spite of its singularity problems, the extended Jacobian method has been used for enhancing performance via various criteria, such as manipulability and obstacle avoidance (Baillieul, 1986). This effort does not provide an adequate test since Baillieul only demonstrated obstacle avoidance for a planar robot with three degrees of freedom performing a very simple positioning task .

2.3.4 Neural Nets

Applying artificial neural nets to the inverse kinematics problem seems natural. Humans and animals solve the inverse kinematics problem with ease. A snake, for instance, is a creature with a very small brain, yet it is able to solve the inverse kinematics problem in real-time while incorporating obstacle avoidance and other high-level criteria. Unfortunately, applications of artificial neural nets to the inverse kinematics problem have had limited success, even with simple robot geometries. Nonetheless, there are several advantages that a successful artificial neural inverse kinematics method would possess. Before being applied to a problem, artificial neural nets must go through an initial training phase.

Much as a baby learns hand-eye coordination, a neural net must be trained to solve the inverse kinematics problem. There are several ways of training the artificial neural net. If the actual robot is available, it can provide the artificial neural net with the joint angles that correspond to the desired end-effector position. If the actual robot is not available, another inverse kinematics algorithm could provide the artificial neural net with the actual joint angles that correspond to the desired end-effector position. In both of these cases the artificial neural net is trained with the correct set of joint angles. If no means of generating the proper solutions are available, then the artificial neural net can only be trained in a trial and error fashion. In this case the training period will be much longer than if the correct solutions were available.

Artificial neural nets as applied to the inverse kinematics problem typically use error back propagation in the training phase (Pourboghrat, 1989).

Through error back propagation, multilayer artificial neural nets learn the inverse kinematics mapping experientially. For each training case the synaptic weights and thresholds are adjusted to minimize the error between the trial and actual solution. The process of generating trial solutions, computing the error, and adjusting the synaptic weights and thresholds continues until all the inverse kinematics solutions from the training set are learned within an acceptable error.

There are several advantages that an artificial neural net solution to the generalized inverse kinematics problem would possess. The main advantage is that an artificial neural net is non-algorithmic and robot independent. That is, it may be possible that the same artificial neural net could be trained the inverse kinematics relationship for any robot. This is certainly attractive, especially if the training phase could be accomplished in a reasonable amount of time. Another advantage is that the artificial neural nets can optimize performance criteria. In artificial neural net jargon the redundant inverse kinematics problem is known as a "one-to-many" mapping problem.

The main disadvantage with applying artificial neural nets to the inverse kinematics problem is their inability to guarantee the accuracy of the solution. Limited accuracy is also apparent in natural neural nets. Imagine a person trying to grind piece of steel to within a .001 inch tolerance using a hand-held grinder. It's just not possible. Researchers attempting to alleviate the problem with accuracy have combined the artificial neural net with iterative numerical methods, such as Newton-Raphson (Egeland et al., 1991). The artificial neural net generates an initial solution, while the Newton-Raphson method refines it to the desired level of accuracy.

A possible problem is that artificial neural nets are almost invariably demonstrated only for the planar case. Presumably a much larger net would be required for the vastly more complex spatial case. This large artificial neural net would require a large amount of machine memory and might be too slow for practical application. However, specialized artificial neural net electronic hardware may alleviate these problems.

2.3.5 Genetic Algorithms

Genetic algorithms are search procedures based on the principles of genetics and natural selection. These algorithms possess the two qualities required of a solution to the inverse kinematics problem: they can be applied to any robot geometry because they treat all problems as a "black box" and they allow the consideration of an unlimited number of performance criteria. These criteria are assembled as a single performance index called "fitness" in genetic algorithm jargon. The main components of a genetic algorithm are: population, reproduction, crossover, and mutation (Parker and Goldberg, 1989).

As applied to the inverse kinematics problem, genetic algorithms maintain a number of joint angle sets called the population. Binary numbers represent the joint angles. Each bit in the binary number depicts a bit of genetic information. The joint angle sets are evaluated according to their fitness for solving the inverse kinematics problem. For instance, the Cartesian error is a measure of fitness. The fittest (least error) joint angle sets are given the highest probability of selection for mating and its associated genetic action.

The genetic action takes place during crossover. The fittest sets of joint angles have respective bits within their binary encoded numbers swapped. This results in new sets of joint angles. In essence, these sets are the "offspring" of the previous sets.

Finally, mutation is the random changing of bits from zero to one or from one to zero. Mutation is typically used sparingly and is of secondary importance to reproduction and crossover.

The process of choosing joint angle sets, swapping bits, and then evaluating fitness continues until, presumably, a solution is found. This is where the problem occurs. The population may stabilize before reaching a solution. Typical errors are on the order of .25 inches for "industrial size" robots. This positioning error is altogether unacceptable and currently precludes genetic algorithms from being a practical method of solving the inverse kinematics problem. There may, however, be technological advances (specialized hardware, etc.) which, combined with the inherent generality of genetic algorithms, would warrant their future development as a solution to the generalized inverse kinematics problem.

Table 2.3 Some positive and negative qualities of several inverse kinematics methods for redundant robots.

Method	**Positive Qualities**	**Negative Qualities**
pseudoinverse	simple formulation	not conservative singularities unreliable

extended Jacobian	can incorporate criteria	not conservative geometric singularities unreliable algorithmic singularities
neural nets	seem to work for humans general with respect to geometry general with respect to performance criteria	limited accuracy
genetic algorithms	general with respect to geometry general with respect to performance criteria	limited accuracy

2.3.6 Performance Criteria

Of all the research supporting generalized inverse kinematics, none is more important than developing of performance criteria. Performance criteria are based on kinematic and dynamic models of the robot. The generalized inverse kinematics strategy uses these criteria to evaluate the performance of the robot and make intelligent decisions as to which kinematic configuration will be most desirable for executing the specified task. Many different performance criteria are presented in the literature. While some of these performance criteria are valuable, their development is not thorough enough to support generalized inverse kinematics. Fortunately, the University of Texas Robotics Research Group has, through two doctoral dissertations and several masters theses, systematically and rigorously developed a mathematical formulation of at least thirty performance criteria. These performance criteria are categorized according to the order of the geometry and energy domain represented. Normalization procedures have been developed which will enable a generalized inverse kinematics strategy to compare and consider many different performance criteria which may originally be of widely varying magnitude.

Of the performance criteria presented in the literature, the joint range availability is the criterion of choice when controlling an actual robot. The Robotics Research Corporation patented their inverse kinematics method incorporating joint range availability. The joint range availability is used with real robots because real robots have real joint travel limits that cannot be ignored. As it is typically formulated, the joint range availability, JRA, is a simple criterion easily calculated.

$$\mathrm{JRA} = \sum_{i=1}^{n}\left[\frac{\left(\Phi_i - \tilde{\Phi}_i\right)^2}{\Phi_{i\,\max}^2}\right] \tag{2.21}$$

In this formulation, Φ_i is the joint displacement, $\tilde{\Phi}_i$ is the mid-range displacement and $\Phi_{i\,\max}$ is the displacement at the joint limit. Minimizing this criterion keeps the joints as near as possible to the midpoints of their travel. While observing joint limit constraints is mandatory, seeking to keep the joints near their center positions is not necessarily associated with enhanced performance. For a given task, it is only requisite that the joint limit constraints not be violated. Other performance criteria, such as those associated with forces and deflections, might indicate that superior performance is available if one or more joints are near, but not exceeding their limits.

The proximity to a singularity is a common performance criteria discussed in the literature (Burdick, 1991). The Jacobian matrix is rank deficient at singularities. As most inverse kinematics techniques for redundant robots invert the Jacobian (one way or another), it is obvious why singularity avoidance is of paramount importance to these algorithms. Several criteria for measuring the proximity of a singularity have been identified. The transmissibility of the Jacobian, ω, is a common one.

$$\omega = \sqrt{|\mathbf{J}\mathbf{J}^{\mathrm{T}}|} \tag{2.22}$$

Another performance criteria for detecting singularities is the condition number of the Jacobian, cond($\mathbf{J}$). As shown in Equation 2.23, the condition number is the ratio of maximum singular value, $\sigma_{\max}$, to the minimum singular value, $\sigma_{\min}$.

$$\mathrm{cond}(\mathbf{J}) = \frac{\sigma_{\max}}{\sigma_{\min}} \tag{2.23}$$

Researchers suggest that the best and most direct measure of a singularity is the minimum singular value of the Jacobian. This raises a fundamental question. Are singularity avoidance criteria measures of the robot's performance or are they measures of the inverse kinematics algorithm's performance? Even for robots with six degrees of freedom, which lose end-effector dexterity at singularities, there are still feasible directions for moving the end-effector. For robots with one degree of redundancy, some singular positions represent areas of unacceptably high demands on joint speeds. Other singular positions do not diminish the capabilities of the robot. As the degree of redundancy increases, the relationship between singularities and robot performance becomes even less clear. Consider the hyper-redundant robot shown in Figure 6.20. Each time one of the pivot joints moves through the center of its travel, the axes of the roll joints on either side of the pivot are collinear and the Jacobian is rank-deficient. These singularities, however, represent no loss in performance capabilities for the robot. For this robot, avoiding singularities results in extremely diminished performance capabilities. All things considered, the relationship between singularity avoidance performance criteria and the actual performance of the robot are dubious at best.

Obstacle avoidance is clearly of paramount importance with robots. Obstacle avoidance refers to moving the robot while avoiding collisions with objects in the environment. The objects include the robot itself, fixed objects in the environment, moving objects in the environment, and also other robots operating in the same workspace. The cost of an industrial robot colliding with an obstacle in the environment could be very high, both in terms of repair to the robot and its environment as well as the cost of lost productivity due to downtime. The consequences of a collision in a more sensitive environment – such as in space, nuclear, or military applications – could be enormous.

Conceptually, the obstacle avoidance problem is easy to understand. The mathematical formulation of the problem is, however, much more difficult. For instance, actual physical objects have edges and corners which are difficult to describe with continuous mathematical functions. There are numerous other complicating factors too extensive to be covered here. The following discussion merely describes several obstacle avoidance formulations used in conjunction with solving the inverse kinematics problem.

A problem similar to obstacle avoidance is target acquisition. This is basically the inverse of the obstacle avoidance. Rather than repelling the robot, as would an obstacle, the target attracts the robot. This formulation may be used to draw the robot's end-effector towards its desired placement.

Wellman and Tesar (1991) developed an obstacle avoidance technique using intersecting volumes. These volumes were placed around the robot and any obstacles in a known world model. Intersections of the volumes were used to calculate forces that would be fed back to the robot's operator so that the operator would not drive the robot or its payload into a collision. The main strength of this method is that is computational complexity matched current computational capablilties.

Maciejewski and Klein (1988) formulated the obstacle avoidance problem in terms of an obstacle avoidance point. They maximized the distance between this point and the obstacle using the pseudoinverse. A problem with the pseudoinverse, as applied to obstacle avoidance, concerns conservative motion. Since a robot employing the pseudoinverse typically drifts in joint space, it is not enough to monitor the robot the first time it performs a task among obstacles. Each time the robot performs the task it will likely move through different paths in joint space. One of these paths may lead to a singularity or collision.

Das, Slotine and Sheridan (1988) formulated the obstacle avoidance problem using what they named the Jacobian Transpose method. This method employs the transpose of the Jacobian matrix rather than the pseudoinverse. The method is quite interesting. Physically, it can be thought of as the application of a force that directs the tip of the robot towards the desired end-effector solution.

The problem is then formulated as a control loop driving the end-effector towards the desired displacement. The resulting joint displacements represent the solution to the inverse kinematics problem. For a redundant robot, other forces can be added to direct the robot away from obstacles. Das, Slotine and Sheridan simulated this algorithm for a planar robot with ten rotational degrees of freedom and the results seem to be very good. There is, however, quite an increase in complexity when moving from the planar to the spatial case. It will be interesting to follow this work and see if it can be successfully extended to spatial robots

The previously discussed performance criteria are dispersed in the literature and in no way represent a systematic development. Typically, these criteria are formulated for ease of computation or for convenience as related to their associated inverse kinematics methods.

In contrast, the University of Texas Robotics Research Group put forth a concerted and systematic effort developing a set of performance criteria physically relevant to the operation of robots. Cleary and Tesar (1990) arranged these criteria into the following categories: geometric, inertial, kinetic energy distribution, and system compliance. Bevill and Tesar (1990) developed a procedure for normalizing these criteria.

The geometric performance criteria are further subdivided into first and second order groups. The first order criteria are based on the Jacobian matrix while the second order criteria are based on the Hessian array. These criteria are task independent and based only on the geometry of the robot. Therefore, they need not be reformulated for different tasks.

The first five entries in Table 2.4 list the first order geometric criteria. The singularity detection criterion is simply the minimum singular value of the Jacobian and measures how close the Jacobian is to rank deficiency. As has been discussed, this criterion is of limited value for robots with a high degree of redundancy.

Table 2.4 Geometric performance criteria developed at U. T. Austin.

Criterion	**Symbol**
singularity detection	η_σ
dexterity	η_δ
velocity transmission	$\eta_{v\chi}$
force and torque transmission	$\eta_{\tau\chi}$
Jacobian Frobenius Norm	η_Σ
rate of change of singularity detection criterion	$\eta_{\Delta\sigma}$
rate of change of dexterity criterion	$\eta_{\Delta\delta}$
end-effector velocity-induced joint acceleration	$\eta_{\phi\alpha}$
joint velocity-induced end-effector acceleration	$\eta_{\upsilon\alpha}$

The dexterity criterion, η_δ in Equation (2.24), is the ratio of the minimum and the maximum singular values of the Jacobian and measures the robot's dexterity at any given set of joint displacements.

$$\eta_\delta = \frac{\sigma_{\max}}{\sigma_{\min}} \tag{2.24}$$

This is an important criterion for robots with a limited degree of redundancy but, as with the singularity detection criterion, becomes less useful as the degree of redundancy of robot increases.

The velocity transmission criteria, η_{vx} in Equation (2.25), measures the robot's ability to move in a desired direction (the x direction in this case).

$$\eta_{vx} = \left\{ \sum_{\mathrm{i}=1}^{\mathrm{M}} \left[\frac{1}{\sigma_{\mathrm{i}}} \left(\dot{\mathbf{u}}^{\mathrm{e}} \right)^{\mathrm{T}} \mathbf{h}_{\mathrm{i}} \right]^{2} \right\}^{-\frac{1}{2}} \tag{2.25}$$

In this equation, M is the number of degrees of freedom at the end-effector, σ_{i} is the i-th singular value of the Jacobian, $\dot{\mathbf{u}}^{\mathrm{e}}$ is the desired end-effector motion, and $\mathbf{h}_{\mathrm{i}}$ is the i-th end-effector space singular vector. This criterion applies equally well to redundant and hyper-redundant robots. Similar to the velocity transmission criteria, the force and torque transmission criteria, $\eta_{\tau x}$ in Equation (2.26), measures the robot's ability to produce a force or torque in a desired direction (the x direction in this case).

$$\eta_{\tau x} = \left\{ \sum_{\mathrm{i}=1}^{\mathrm{M}} \left[\sigma_{\mathrm{i}} \left(L^{\mathrm{e}} \right)^{\mathrm{T}} \mathbf{h}_{\mathrm{i}} \right]^{2} \right\}^{-\frac{1}{2}} \tag{2.26}$$

The terms in this equation are the same as those in (2.25) with the exception that L^{e} is the direction of the force at the end-effector. This criterion seems to be especially well-suited for redundant robots which can use self-motion so as to best apply forces and torques in a desired direction. This capability is of great value during insertion or forming tasks.

The Jacobian matrix Frobenius norm, η_{Σ} in Equation (2.27), also measures velocity, force, and torque transmission capabilities.

$$\eta_{\Sigma} = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{M}} \sigma_{\mathrm{i}}^{2}} \tag{2.27}$$

As noted by Van Doren and Tesar (1992), this criterion is less accurate and does not exhibit as much variation over the workspace as the previously described transmission criteria. This diminishes the value of the Jacobian matrix Frobenius norm in a decision making framework.

The final four listings in Table 2.4 are the second order geometric performance criteria. These criteria are derived primarily from the Hessian array.

The rate of change of the singularity detection criterion, $\eta_{\Delta\sigma}$ in Equation (2.28), indicates whether the robot is moving towards or away from a singularity.

$$\eta_{\Delta\sigma} = \sqrt{\sum_{i=1}^{n}\left(\frac{\partial\sigma_{\min}}{\partial\Phi_i}\right)^2} \tag{2.28}$$

The summation is performed over n, which is the number of the robot's degrees of freedom. The rate of change of the dexterity criterion, $\eta_{\Delta\delta}$ in Equation (2.29), indicates how rapidly the robot's dexterity is increasing or decreasing.

$$\eta_{\Delta\delta} = \sum_{i=1}^{n}\left(\frac{\partial\eta_\delta}{\partial\Phi_i}\right)^2 \tag{2.29}$$

As with their first order counterparts, the usefulness of the second order singularity and dexterity criteria decreases as the redundancy of the robot increases.

Of the final two second order geometric criteria, the end-effector velocity-induced joint acceleration, $\eta_{\Phi\alpha}$ in Equation (2.30), seems to be the most important.

$$\eta_{\Phi\alpha} = \frac{\sum_{i=1}^{n}\lambda_{i\,\max}}{n} \tag{2.30}$$

In this equation, $\lambda_{i\,max}$ is the maximum eigenvalue for the Hessian array. This criterion relates the joint accelerations to a constant end-effector velocity. It is important because many robotic applications, such as spray painting or welding, require constant velocities. It is also important because robot actuators have a finite ability to produce joint-level accelerations. Van Doren and Tesar (1992) show that for an example task this criterion varies considerably, which in turn indicates that it will be useful in a decision making framework.

Table 2.5 lists the force/torque performance criteria. These criteria are based on dynamic models of forces and torques within the robot and are essential to the intelligent design and application of robots. The first four criteria deal with actuator torques. Actuator torque demands are a fundamental consideration, as all robot actuators have a finite ability to produce torques.

Table 2.5 Force/torque performance criteria developed at the U. T. Austin.

Criterion	**Symbol**
dynamic coupling	η_{ζ}
actuator torque	
equivalent end-effector forces	$\eta_{\upsilon\tau}$
end-effector space actuator torque	$\eta_{\upsilon\phi\tau}$
rate of change of actuator torque criterion	
rate of change of equivalent end-effector forces criterion	$\eta_{\Delta\upsilon\tau}$
rate of change end-effector space actuator torque criterion	
velocity-induced actuator torque	
velocity-induced equivalent end-effector forces	$\eta_{\nu\upsilon\tau}$
end-effector space velocity-induced equivalent actuator torque	$\eta_{\nu\upsilon\phi\tau}$
GH norm	η_{γ}

The next three criteria deal with the rate of change of torques. The rate of change of the actuator torque criterion, $\eta_{\Delta\tau}$ in Equation (2.31), measures how fast the robot can respond to torque and force demands.

$$\eta_{\Delta\tau} = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial \lambda_{max}}{\partial \Phi_i} \right)^2} \tag{2.31}$$

This is an especially important criterion because larger actuators or higher gear ratios can supply more torque, but both will likely slow the overall response of the robot to external disturbances. Consideration of the basic torque demands as well as its rate of change could allow the intelligent allocation of the robot's resources for truly enhanced operation.

Bevill and Tesar (1990) developed a performance measure based on the product of the Jacobian matrix and Hessian array that they named the GH norm. The name reflects their physical interpretation of the Jacobian matrix as a collection of gain elements and the Hessian array as a composition of higher order geometric properties. Van Doren and Tesar (1992) then formulated this performance measure as a single criterion. It is shown as η_γ in Equation (2.32).

$$\eta_\gamma = \frac{\sum_{i=1}^{n} \lambda_{i\,GH\,max}}{n} \tag{2.32}$$

In this equation, $\lambda_{i\,GH\,max}$ is the maximum eigenvalue of the symmetric portion of each plane in the Jacobian and Hessian product. This criterion is a mathematical description of torque shock and the relative demand for responsiveness from the associated actuators based on higher order geometric properties.

Table 2.6 lists the kinetic energy performance criteria. Minimizing kinetic energy is important for high-speed precision operation. The kinetic energy criterion addresses high-level issues represented in relatively simply formulations.

Table 2.6 Kinetic energy performance criteria developed at U. T. Austin.

Performance Criterion	**Symbol**
joint space kinetic energy	η_{κ}
end-effector space kinetic energy	$\eta_{\upsilon\kappa}$
rate of change of joint space kinetic energy	$\eta_{\Delta\kappa}$
rate of change of end-effector space kinetic energy	$\eta_{\Delta\upsilon\kappa}$
kinetic energy partition values	$\eta_{\kappa\pi}$

The kinetic energy, KE, of a serial robot may be formulated in a straightforward manner.

$$\mathrm{KE} = \frac{1}{2}\dot{\Phi}^{\mathrm{T}}\left[\mathbf{I}^{*}_{\Phi\Phi}\right]\dot{\Phi} \tag{2.33}$$

$\mathbf{I}^{*}_{\Phi\Phi}$ is the effective inertia matrix. This equation represents the entire kinetic energy content of the robot. In this formulation, the kinetic energy value is dependent upon the joint speeds and is therefore dependent upon the task being performed. Van Doren and Tesar (1992) formulated a task-independent kinetic energy criterion. It is shown as η_{κ} in Equation (2.34).

$$\eta_{\kappa} = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} \lambda_{\mathrm{i}}^{2}} \tag{2.34}$$

The λ_{i}'s are the eigenvalues of the effective inertia matrix.

The rate of change of the kinetic energy criterion is also important. This criterion, $\eta_{\Delta\kappa}$, is formulated as shown in Equation (2.35).

$$\eta_{\Delta\kappa} = \sqrt{\sum_{i=1}^{n}\left(\frac{\partial \eta_{\kappa}}{\partial \Phi_i}\right)^2} \tag{2.35}$$

Large changes in kinetic energy correspond to very large demands on actuator power. Very rapid changes in the kinetic energy represent shocks to the robot.

Table 2.7 lists the compliance performance criteria. The compliance criteria describe the robot's ability to perform precision operations under load. They also correspond to the vibratory modes of the robot. As pointed out by Ambrose and Tesar (1992), the compliance of a robot's joints is typically one order of magnitude greater than the compliance of its links.

Table 2.7 Compliance criteria developed at U. T. Austin.

Criterion	**Symbol**
oscillation	η_{ω}
system stiffness	η_{φ}
system potential energy	η_{ρ}
potential energy partition values	$\eta_{\rho\pi}$

Of the compliance criteria, the potential energy partition values, are particularly important. The potential energy partition values measure the distribution of compliance energy and how it changes as the robot moves. An unusually high compliance energy content in any part of the robot indicates a problem with the robot's design. Rapid changes in compliance energy indicate large local forces, which correspond to large actuator demands and decreased precision.

Using these criteria to make intelligent decisions and enhance the robot's performance is the goal of generalized inverse kinematics. An intelligent decision making process must compare the contributions of each performance criteria and consider the overall performance of the system. The decision making process must use scaling factors to make fair comparisons leading to intelligent decisions.

Bevill and Tesar (1990) suggest using average values calculated over the robot's entire workspace as scaling factors. This approach gives a global measure of the performance criteria's magnitude, though it is computationally intensive. For instance, calculating criteria values for a robot with 7 joints, each having a range of 180 degrees, at a resolution of 10 degrees requires 18^7 or approximately 600×10^6 criteria calculations. Although this represents a significant computational burden, the calculations must only be performed once per robot. Van Doren and Tesar (1992) propose another normalization and scaling procedure that is less of a computational burden. This scaling approach is based on velocity and torque limitations at the joint level.

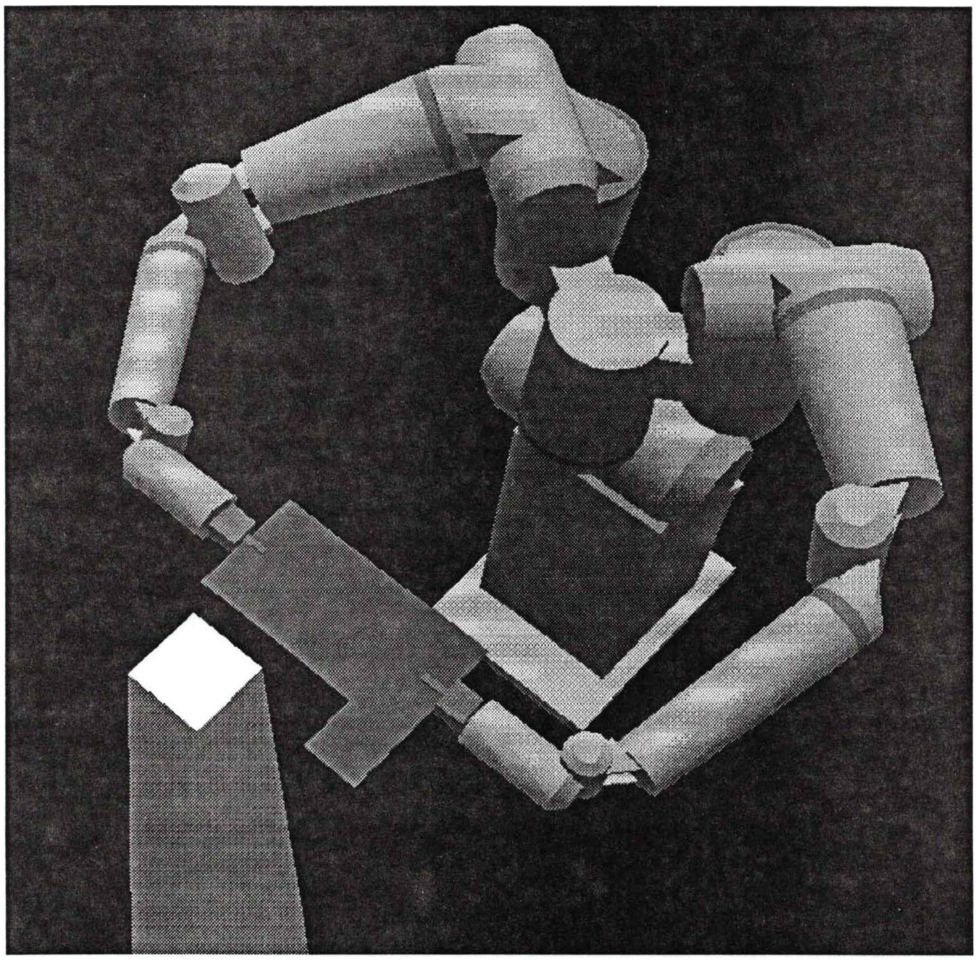

Figure 2.1 An advanced dual-arm robot with 17 total degrees of freedom handlin a thin material.